#3

Docket No. 826.1718

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
	)	
Yoshitake SHINKAI, et al.	)	
	)	Group Art Unit: Unassigned
Serial No.: To be assigned	)	
	)	Examiner: Unassigned
Filed: March 26, 2001	)	
	)	
For: FILE REPLICATION SYSTEM, REPLICATION CONTROL METHOD, AND STORAGE MEDIUM	)	

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. §1.55

Assistant Commissioner for Patents
Washington, D.C. 2023l

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-126797
Filed: April 27, 2000.

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 26, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE

JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: April 27, 2000

Application Number: Patent Application No. 2000-126797

Applicant(s): FUJITSU LIMITED

December 22, 2000

Commissioner,

Patent Office Kozo OIKAWA

Certificate No. 2000-3105863

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 ４月２７日

出 願 番 号
Application Number: 特願２０００－１２６７９７

出 願 人
Applicant (s): 富士通株式会社

２０００年１２月２２日

特 許 庁 長 官
Commissioner,
Patent Office 及 川 耕 造

出証番号 出証特２０００－３１０５８６３

【書類名】　特許願

【整理番号】　9952124

【提出日】　平成12年 4月27日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 12/00

【発明の名称】　ファイルレプリケーションシステム、ファイルレプリケーション制御方法及び記憶媒体

【請求項の数】　22

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　新開　慶武

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　吉沢　直美

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　塩沢　賢輔

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【代理人】

【識別番号】　100074099

【住所又は居所】　東京都千代田区二番町８番地２０　二番町ビル３Ｆ

【弁理士】

【氏名又は名称】　大菅　義之

【電話番号】　03-3238-0031

【選任した代理人】

【識別番号】　100067987

【住所又は居所】　神奈川県横浜市鶴見区北寺尾７－２５－２８－５０３

【弁理士】

【氏名又は名称】　久木元　彰

【電話番号】　045-573-3683

【手数料の表示】

【予納台帳番号】　012542

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9705047

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　ファイルレプリケーションシステム、ファイルレプリケーション制御方法及び記憶媒体

【特許請求の範囲】

【請求項１】　複数のノードがネットワークに接続され、該各ノード上に共用ファイルを配置するファイルシステムにおいて、

前記複数のノード内の１つである第１のノードは、

前記共用ファイルに対するアクセス要求が生じた時、前記複数のノード内の１つである第２のノードに該共用ファイルに対するアクセスの許可を求める第１のトークン管理手段と、

自ノード内で生じた共用ファイルに対するアクセスを受け付け、該アクセス要求に対し前記第１のトークン管理手段に前記アクセスの許可獲得を依頼し、該許可が得られない時、前記共用ファイルに対する更新許可を持つノードに該共用ファイルへのアクセス処理を依頼するＩＯ要求インタセプト手段と、

を備え、

前記第２のノードは、

他ノードからの共用ファイルに対するアクセスの許可要求に対し、別のノードに該共用ファイルに対する更新許可を与えている時、該アクセス許可要求に対する応答として該更新許可を与えているノードを通知する第２のトークン管理手段を備えることを特徴とするファイルシステム。

【請求項２】　他のノードとネットワークによって接続され、該他のノードとの共用ファイルを保持するノードにおいて、

前記共用ファイルに対するアクセス要求を管理するトークン管理手段と、

自ノード内で生じた共用ファイルに対するアクセス要求に対し、前記トークン管理手段に該共有ファイルへのアクセス許可を求めるＩＯ要求インタセプト手段と、

を備え、

前記トークン管理手段は、前記ＩＯ要求インタセプトからのアクセス要求に対し、既に他のノードが前記共用ファイルに対する更新許可を保持する時、該更新

許可を保持するノードを前記ＩＯ要求インタセプト手段に通知し、前記ＩＯ要求インタセプト手段は、前記アクセス許可が得られない時、該更新許可を保持するノードに前記共用ファイルへのアクセス処理を依頼することを特徴とするノード。

【請求項３】　新規参入時に自ノードの保持する共用ファイルのデータの復元処理を行う系構成管理手段を更に備え、前記ファイルの復元処理中に、自ノード内で前記共用ファイルに対するアクセス要求が生じた時、前記ＩＯ要求インタセプト手段は、前記共用ファイルを共用している他のノードにアクセス処理を依頼することを特徴とする請求項２に記載のノード。

【請求項４】　前記共用ファイルへの更新時に更新内容を他の更新との依存関係を示す情報と共に他のノードへ伝播する変更データ通知手段と、

前記依存関係を示す情報に基づいて、更新の順序性を保証しつつ前記更新内容を前記共用ファイルに反映させる受信データ処理手段を更に備えることを特徴とする請求項２又は３に記載のノード。

【請求項５】　１乃至複数の共用ファイル毎に更新内容の伝播方式についての情報を保持する系状態情報保持手段を更に備え、前記変更データ通知手段は、前記形状情報保持手段内の情報に基づいて前記更新内容を伝播することを特徴とする請求項４に記載のノード。

【請求項６】　前記伝播方式は前記共用ファイルを共用する全てのノードに前記更新内容が伝播されるのを保証する同期方式、前記前記共用ファイルを共用する半数のノードに前記更新内容が伝播されるのを保証する半同期方式、及び前記共用ファイルを共用するノードへの前記更新内容の伝播を確認しない非同期方式のいずれか１つであることを特徴とする請求項５に記載のノード。

【請求項７】　前記系状態情報保持手段は、前記１乃至複数の共用ファイル毎に該共用ファイルを共用するノードについての情報情報をも保持することを特徴とする請求項４乃至６のいずれか１に記載のノード。

【請求項８】　複数のノードがネットワークに接続され、該ノードが共用ファイルを共用する構成のシステムにおけるファイルレプリケーション制御方法であって、

前記共用ファイルに対するアクセスを行うアクセス要求ノードは、ノードが前記共用ファイルに対する最新のデータを自己が保持する時、自己の共用ファイルにアクセスし、

前記最新のデータを他ノードが保持する時、前記共用ファイルに対するアクセスを該他ノードに依頼することを特徴とするファイルレプリケーション制御方法。

【請求項９】　前記共用ファイルへの更新許可は１つのノードにのみ与えられ、前記アクセス要求ノードは共用ファイルにアクセスする時に、他ノードが前記共用ファイルへの更新許可を保持している時、該更新許可を保持しているノードに前記共用ファイルへのアクセス処理を依頼することを特徴とする請求項８に記載のファイルレプリケーション制御方法。

【請求項１０】　前記共用ファイルへの更新を行ったノードは、更新内容を他ノードに非同期で伝播し、

前記更新内容が伝播中に他ノードで生じた共用ファイルへのアクセス要求を前記更新を行ったノードが処理することを特徴とする請求項８又は９に記載のファイルレプリケーション制御方法。

【請求項１１】　前記更新許可を保持しているノードは、自己の更新が依存する更新が全ノードに伝わった後、該更新許可の解放を行うことを特徴とする請求項乃至１０のいずれか１つに記載のファイルレプリケーション制御方法。

【請求項１２】　前記共用ファイルへの更新内容は順序性を保証して反映されることを特徴とする請求項８乃至１１のいずれか１つに記載のファイルレプリケーション制御方法。

【請求項１３】　他の更新との順序関係を示す依存情報を前記更新内容と共に他ノードに伝播することを特徴とする請求項１２に記載のファイルレプリケーション制御方法。

【請求項１４】　前記更新内容を受信したノードは、前記依存情報に基づき、該更新内容に先行する更新内容を受信した後で、該更新内容を自己の共用ファイルへ反映させることを特徴とする請求項１３に記載のファイルレプリケーション制御方法。

【請求項１５】　前記共用ファイルへの更新内容の他ノードへの伝播の方式を１乃至複数の前記共有ファイル単位で指定することを特徴とする請求項９乃至１４のいずれか１つに記載のファイルレプリケーション制御方法。

【請求項１６】　前記共用ファイルへの更新内容を伝播するノードを１乃至複数の前記共有ファイル単位で指定することを特徴とする請求項９乃至１５のいずれか１つに記載のファイルレプリケーション制御方法。

【請求項１７】　新規参入時に自ノードの保持する共用ファイルのデータの復元処理を行い、該復元処理完了前にユーザプログラムを稼動させることを特徴とする請求項９乃至１６のいずれか１つに記載のファイルレプリケーション制御方法。

【請求項１８】　前記復元処理によるデータの送信は、前記共用ファイルへの更新要求に対する処理と順序性を保証して行われることを特徴とする請求項１７に記載のファイルレプリケーション制御方法。

【請求項１９】　前記復元処理完了前に生じた前記共用ファイルへのアクセス要求に対する処理を、前記共用ファイルを共用している他のノードに依頼することを特徴とする請求項１７又は１８に記載のファイルレプリケーション制御方法。

【請求項２０】　共用ファイルに対する処理を、該共用ファイルを共用する他ノードと同期して停止する整然停止を行ったとき、整然停止を行ったノードは該整然停止を行ったことを記憶し、該共用ファイルへの処理を再開する際、他ノードと同期して再開することにより該共用ファイルに対するデータを復元処理を行わないことを特徴とする請求項９乃至１９に記載のファイルレプリケーション制御方法。

【請求項２１】　複数のノードがネットワークに接続される構成のシステムにおけるファイルレプリケーション方法であって、

第１のノードはファイルにアクセスする時に、トークン獲得を要求し、

前記要求に対し前記第１のノードがトークンを獲得できない時は該トークンを

保持している第２のノードを前記第１のノードに通知し、

前記第１のノードは、前記獲得できない事を通知された時、前記第２のノードに前記ファイルアクセスを依頼する

ことを特徴とするファイルレプリケーション方法。

【請求項２２】　ネットワークにより他ノードと接続されるノードを構成するコンピュータにより使用された時、

前記各ノードが共用する共用ファイルに対するアクセスを行う時、前記共用ファイルに対する最新のデータを自己が保持するときは自己の共用ファイルにアクセスし、

前記最新のデータを他ノードが保持するときは、前記共用ファイルに対するアクセスを該他ノードに依頼することを前記コンピュータに行わせるためのプログラムを記憶した前記コンピュータが読み出し可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、複数のコンピュータ間にファイルの複製を動的に配置し、負荷分散をはかり性能向上を実現すると共に信頼性を向上させるファイルレプリケーション技術に関する。

【０００２】

【従来の技術】

従来、ネットワークで接続されている複数の計算機システム（ノード）間に同一のデータを動的に配置し、信頼性を向上させる方式として、ファイルレプリケーション技術が知られている。

【０００３】

ファイルレプリケーションでは、あるノード上のファイルが更新されたタイミングでファイルの更新内容を知り、予め定義された他のノード群に変更されたデータのみを伝播してファイルを更新させる。

【０００４】

更新内容の伝播の仕方としては、更新命令の完了がユーザプログラムに通知さ

れた時点で、他ノードへの伝播が完了していることを保証する同期型と、システム内に変更内容を蓄積し、適当なタイミングで他ノードへ伝播する非同期型の伝播が知られている。このうち非同期型の場合は、応答性が良く処理性能は高いが、更新命令の完了がユーザプログラムに通知された時点で、他ノードに更新内容が伝わっている保証はない。

【０００５】

一方従来のファイルレプリケーション方式では、各ノードが保持するデータの同一性あるいは一貫性が保証されていないため以下の問題が発生する。

まず非同期型の場合、複数のノードが関連する異なるファイルを順に更新した場合、更新の伝播の順序性が保証されない。その為、参照だけを行うノードだけからみていても、新旧入り交じった一貫性の無いデータが見えてしまうという本質的な欠点を抱えている。

【０００６】

また複数のノードが同じファイルをほぼ同時に更新（実時間ではかなりずれている場合を含む）すると、各ノードが異なったデータを保持することになり、結果的にファイルが破壊される。

【０００７】

このデータの破壊については、非同期伝播を用いた場合のみならず同期型伝播を採用してた場合においても、２つのノードがほぼ同じ時刻に更新した場合にはファイルが破壊されることがある。例えば同じファイルの重複する領域をノードＡとノードＢの２つのノードがほぼ同じ時刻に更新した場合、ノードＡとＢが異なるデータを保持する結果になることがある。この場合、その後の処理においては各ノードは自身が保持する互いに異なるデータに基づいて処理を続行するのことになるので、最終的には矛盾した処理がノードＡとノードＢで行われることになる。

【０００８】

この為、従来のファイルレプリケーション方式では、スタティックに決まる１つのノードにのみファイルの更新処理を許し、他ノードに対しては参照のみを許すという制約を与えていた。この方式によるものとしては、特開平９－９１１８

５号「分散コンピューティングシステム」がある。ここで提案されている方式では、自ノードのデータを更新あるいは参照できることを示すWrite トークンと、自ノードのデータを参照のみできることを示すReadトークンを用意し、Write トークン保持ノードが存在する時には、他のノードのいずれもRead/Writeトークンを保持していないように制御し、かつ更新要求を全て同期型で行うことで、同時更新に伴う矛盾を解消している。

【０００９】

【発明が解決しようとする課題】

しかし上記公報に開示されている方式では、ファイルの更新が常に同期型で行われるため、応答性の問題を持つ。また、同じファイルを同時にアクセスする複数のノードが存在し、かつその中に１つ以上がファイルの更新を行うものであった場合、自ノードのデータをアクセスする為に必要となるトークンの取り戻し処理をアプリケーションプログラムがＩＯ要求を発行する度に行わなければならず、オーバヘッドが非常に大きくなってしてしまう。

【００１０】

また、この方式を含め従来のレプリケーション方式では、アクセスは常に自ノードが保持しているデータに対して行われることを前提としているため、新規ノードを系に組込む場合、新たに組込まれたノードは関連する全てのファイルのデータを系内の他のノードから自ノードに一括して取込んだ後でないとデータの一貫性が保証されない。この為、新規ノードは系に加わっても直には業務に移れない、新規参入ノードへのデータ取込み中既存系の更新が止まる、即ち長時間通常業務が停止するという欠点を持っていた。

【００１１】

本発明は、最新データを保持しているノードを特定し、そのノードにRead/Write要求を伝播してファイルアクセスを依頼することにより、新規ノード参入時の稼動中業務への影響を最小化することが出来るファイルレプリケーションシステムを提供することを課題とする。

【００１２】

また非同期型の伝播においても複数ノードでの同時更新が可能な高速レプリケ

ーションを実現するファイルレプリケーションシステムを提供することを課題とする。

【００１３】

更に本発明では、非同期に送られる更新要求のファイルへの反映を、Write 要求のノード内順序性を示す更新番号とそのWrite 要求が前提とする他のノードの更新番号からなる依存ベクタを用いて制御することで、系縮退時でもファイル更新の論理的順序性を保証するファイルレプリケーションシステムを提供することを課題とする。

【００１４】

【課題を解決するための手段】

図１は本発明のによるノードの原理図である。

本発明によるノード１は、他のノードとネットワークによって接続され、該他のノードとの共用ファイル６を保持することを前提としており、ＩＯ要求インタセプト部２及びトークン管理手段３を備える。

【００１５】

トークン管理手段３は、上記共用ファイル６に対するアクセス要求を管理する。

ＩＯ要求インタセプト手段２は、自ノード内で生じた共用ファイル６に対するアクセス要求に対し、上記トークン管理手段に該共有ファイルへのアクセス許可を求め、許可が得られると共用ファイル６へアクセスする。

【００１６】

上記トークン管理手段３は、上記ＩＯ要求インタセプトからのアクセス許可に対し、既に他のノードが上記共用ファイルに対する更新許可を保持する時、該更新許可を保持するノードを上記ＩＯ要求インタセプト手段に通知し、上記ＩＯ要求インタセプト手段は、上記アクセス許可が得られない時、該更新許可を保持するノードに上記共用ファイルへのアクセス処理を依頼する。

【００１７】

これにより、各ノード１は共用ファイル６へのアクセスを、最新のデータを保持しているノードのデータに対して行え、また各ノードからは一貫性の有るデー

タが見える。

【００１８】

ノード１は、また上記共用ファイル６への更新時に更新内容を他の更新との依存関係を示す情報と共に他のノードへ伝播する変更データ通知手段４と、上記依存関係を示す情報に基づいて、更新の順序性保証をしつつ上記更新内容を上記共用ファイルに反映させる受信データ処理手段５を更に備える構成とすることも出来る。

【００１９】

この構成により、ファイルの更新内容が更新順と前後して到着しても、共用データ６は、順序性をが保証された更新が行われる。

更にノード１は、新規参入時に自ノードの保持する共用ファイル６のデータの復元処理を行う系構成管理手段を更に備える構成とすることも出来る。この構成の場合、上記ファイルの復元処理中に、自ノード内で上記共用ファイルに対するアクセス要求が生じた時、上記ＩＯ要求インタセプト手段２は、上記共用ファイル６を共用している他のノードにアクセス処理を依頼する。

【００２０】

これの構成により、新規参入したノードは共用ファイルの更新処理の完了を待たずに他の処理に移れる。

【００２１】

【発明の実施の形態】

以下に本発明に於ける一実施形態について図面を参照しながら説明する。

本実施形態のファイルレプリケーションシステムは、複数のノードがネットワークに接続されて系を構成し、系内の各ノードがファイルを共用する構成を前提としている。

【００２２】

まず本実施形態での系の構成について説明する。

図２は本実施形態での系及び系の再構成を説明する図である。

本実施形態で系とは、同一のファイル群（以下各系で共用している１乃至複数のファイル（群）をオブジェクトグループという）を共用しているノードのグル

ープを指す。例えば図２では、オブジェクトグループａ，ｄを共用しているノードＡ、Ｃ、Ｅ及びＦで構成される系ａ、オブジェクトグループｂを共用しているノードＡ、Ｂ、及びＤで構成される系ｂ、オブジェクトグループｃを共用しているノードＧ、Ｈ及びＩで構成される系ｃの３つの系が構成されている。

【００２３】

この系内のノードの内、１つのノードが系内の共用ファイルへアクセスするためのRead／Write トークンを管理している。このトークンを管理するノードには、系を構成する際に、予め決められているノードがなるか、特定の条件、例えば最小ネットワークアドレスを持つものが動的に選ばれる。

【００２４】

また新規のノードが系に加わったり、構成要素となっているノードやネットワークの障害等で系の縮退が生じた時、系の再構成が行われる。例えば、図２の場合系ａではノードＥの障害によりノードＥ及びＦがネットワークから脱落して残りのノードによる系の再構成が行われている。また系ｃでは、ノードＪがJoinコマンドによって新規に系に加わったことにより系の再構成が行われる。この系の再構成の際には、新規参加ノードの共用ファイルの一貫性（consistency ）保証のため等価性回復処理が行われる。

【００２５】

尚ノードの系からの離脱は、障害などによって生じるものの他、離脱を行うノードが系内の他のノードにメッセージを送信して自律的に行うものが有る。

図３は本発明に於けるノード間の基本動作を説明する図である。

【００２６】

図３（ａ）は、オブジェクトグループに対してアクセスする際の、ノード間の処理を示す図で、同図では同一の系にノードＡ～Ｅの５つのノードあり、そのうちノードＡがトークン管理ノードとする。各ノードはユーザプログラムからオブジェクトグループ内のファイルに対するアクセス要求が生じると、ノードＡにRead/Writeトークンの獲得要求を発行する。

【００２７】

これに対しノードＡは、他のノードに既にWrite トークンを渡していなければ

、要求されたトークンを与える。またもし既にWrite を他のノードに渡していれば、トークン獲得失敗通知と共にWrite トークンを保持しているノードを通知する。トークン獲得失敗を通知されたノードは、ファイルへのRead/Write要求をこの通知されたノード対して依頼し、Write トークンを保持しているノードは、ファイルの順序性を保つようにこれらの要求を処理してゆく。同図の場合ノードＢ、ＣがRead要求（参照要求）を、ノードＤがWrite 要求（更新要求）を発行した時点でWrite トークンはＥが保持しているので、ノードＡは各ノードからのトークン獲得要求に対して、獲得失敗と共にノードＥがWrite トークンを保持していることを通知する。これに対して各ノードはノードＥに対してファイルへのRead/Write要求を発行し、ノードＥはファイルの順序性を保持しながらこれらの要求に対しファイルへのRead/Writeを行う。

【００２８】

この様に本発明では、共用ファイルへのアクセス要求が生じたノードに対しWrite トークン保持ノードの通知という形で、その共用ファイルに対する最新のデータを保持するノードが通知される。よって、共用ファイルをアクセスするノードは常に最新のデータに対してアクセスすることが出来る。

【００２９】

また各ノードは、トークンの獲得に失敗してもトークンを獲得できるまで待つことなく処理を続行できる。更に複数のノードによる同一のファイルに対する同時アクセスを可能としている。この為、高い反応性を持つシステムを構築することができる。

【００３０】

またファイルへの変更処理は、他のノードで発生した更新要求に対する処理もWrite トークンを持つ１つのノードが行うので、各ノードからは一貫性のあるデータが見える。

【００３１】

更に同時に生じたアクセス要求を処理する際、それぞれに対するトークンの回収処理を行う必要が無く、オーバヘッドを小さくすることが出来る。

次に本発明に基づいたシステムに於ける系への新規参入時処理については説明

する。

【００３２】

図３（ｂ）は、系に新たに加わったノードの系内の他のノードからとの処理を表す図である。

本発明では、各ノードはデータの最新性を示す情報を保持しており、新規参入ノードはこの情報を比較して、自己が系から離脱している間にデータが更新された時のみ復元処理を行う。また新規ノードはデータの復元処理中に、ユーザプログラムを起動し通常業務に入る。そしてユーザプログラムからファイルへのアクセス要求が発生した場合には、系内の他のノードにRead/Write要求を発行し、ファイルへのアクセス処理を依頼する。図３（ｂ）では、系に新規参入したノードＤはファイルの復元処理の完了を待たずにユーザプログラムを起動して、ファイルの復元処理中にユーザプログラムからオブジェクトグループ内のファイルへのアクセス要求が生じると、このアクセス要求をWrite トークンを保持してるノードＥに依頼している。

【００３３】

この様に本発明では、新規ノードはファイルの復元処理の完了を待たずに、ファイルへのアクセスを行うことが出来るので、系への参入後直ちにユーザプログラムを起動して通常処理を開始することが出来る。

以下に上記基本原理を実現するための一実施形態について図面を参照しながら説明する。

【００３４】

図４は本実施形態の系を構成する複数のノードの内の１つの構成を示すブロック図である。

システム内の複数のディスク装置上に置かれるオブジェクトグループを共用する各ノード１０は、系構成管理部１１、ＩＯ要求インタセプト部１２、トークン管理部１３、変更データ通知部１４及び受信データ処理部１５が配置される構成となっている。これらの各構成要素は、各ノード内でメモリ上に展開されるプログラムによって実現される。また処理速度を得る為、一部をハードウエアにより

実現する構成としてもよい。また、ノード１０のローカルディスク装置１８には、同一の系内で共用している共用ファイル１９及び系構成の為の定義情報である環境定義・状態情報２０を記憶している。

【００３５】

尚これらの構成要素の内、ＩＯ要求インタセプト部１２はオペレーションシステム（ＯＳ）の一部として動作し、ユーザプログラム１７が発行した入出力命令を受取り、ＯＳ内のファイルシステムにこの入出力命令を伝える役割をする。

【００３６】

尚、本実施形態では、ＩＯ要求インタセプト部１２をＯＳのファイルシステム１６と分離した構成としているが、ファイルシステム１６内に含める構成とすることも可能である。また他の構成要素は、ＯＳ内の要素として構成としてもよいし、アプリケーションプログラムとしてＯＳ上に実装する構成としてもよい。

【００３７】

以下、各構成要素について詳細に説明する。

［系構成管理部］

系構成管理部１１は、ノード起動時や系再構成時における系構成状態の監視、対象ファイルや伝播モードについての設定、ノード障害などに伴う系の縮退や新規ノードの参入等系の状態管理、系再構成時の他ノードとの同期（同期回復）、新規参入ノードの初期同期（等価性回復）、ノードの状態の監視及びオペレータとのインタフェース処理を司どる部分である。

【００３８】

また系構成管理部１１は、Joinコマンドにより系に加わりLeave コマンドにより系から脱退するまで、後述する系を構成するノードのノード障害監視処理を行う。

【００３９】

システム立ち上げの一環としてファイルレプリケーションシステムを実現するプログラムが起動されると、まず環境定義・状態ファイルが読み込まれ、複数の対象とするオブジェクトグループに属するファイル群、そのオブジェクトグループを配置するノード群、及び更新データの伝播モードについての情報を得る。

【００４０】

この環境定義・状態ファイルは、各オブジェクトグループ毎に構成された系状態テーブルによって構成されている。

図５は系状態テーブルの構成例を示す図である。

【００４１】

各系状態テーブルは、オブジェクトグループ毎にそのオブジェクトグループの構成等の情報を記録したテーブルである。各系状態テーブルにはそのテーブルに情報が記憶されているオブジェクトグループを識別するオブジェクトグループ番号、系のバージョン番号、自己が前回整然停止したかどうかを表示する整然停止フラグ、系を構成する各ノードを特定する複数のノード番号とそのノードが前回整然停止したかどうかを示すフラグとからなる複数の配列で構成されるノード定義部、このオブジェクトグループに属する各ファイルを特定するオブジェクトグループ定義部及びこのオブジェクトグループに属するファイルの更新データ伝播モード（同期，半同期，非同期：これらの詳細については後述する）を指定する情報によって構成されている。尚「整然停止」とは、例えば正月休み等でサービスを休止する時に、系内のノードが同期を取って同時にそのオブジェクトグループに対する処理を停止する系からの離脱の仕方を指す。

【００４２】

尚図５中の＊部分の情報は、初期値はユーザが設定し、以降系構成管理部１１が必要に応じて変更する情報である。また＊が記されていない部分は、ユーザは設定を行わず系構成管理部１１のみが設定、変更する情報であることを示している。

【００４３】

環境定義・状態情報２０は、複数の系状態テーブルからなる構成であり、複数のオブジェクトグループそれぞれに対して設定を行うことが可能である。よって、オブジェクトグループ毎に異なるノード群や更新データの伝播モードを設定することが出来る。例えば、図２において、ノードＡは、オブジェクトグループａ、ｂ及びｄの３つのオブジェクトグループに対する系状態テーブルを持ち、それぞれに異なったノード群（オブジェクトグループａ及びｄにはノードＣ、Ｄ、Ｅ

及びＦ、オブジェクトグループｂにはノードＢ、Ｃ及びＤ）と転送方式（同期，非同期，半同期）を設定することが出来る。そして、データの重要度に応じて、例えば、最も重要なオブジェクトグループａには同期モード、重要度の低いｃには非同期モード、その中間のオブジェクトグループｂには半同期モード等のそれぞれのオブジェクトグループ毎に異なった設定を行うことが出来る。

【００４４】

環境設定部は、この環境定義・状態情報２０を読み込んで、メモリ上に内部制御表を各オブジェクトグループ毎に展開し、各構成要素にユーザが指定した設定を伝える。

【００４５】

この内部制御表は、ユーザが設定したオブジェクトグループの情報を保持するノードのメモリ上に展開されるテーブルで、例えば、図６の様な構成を取る。

図６の内部制御表は、各オブジェクトグループを特定するオブジェクトグループ番号、更新データのデータ伝播モード（同期，非同期，半同期）、状態フラグ、オブジェクトグループ定義部、ノード定義部、及び更新伝播送信キューと実反映遅延キューのエントリを示すポインタを記録している。このうちオブジェクトグループ定義部は、系状態テーブルのオブジェクトグループ定義部と同様、そのオブジェクトグループに属するファイル群を特定する先頭ファイルパス名の集合を保持しており、この中に特定されたパス名で始まるファイル群がこのオブジェクトグループに属することを示す。またノード定義部内のノード番号とstatusからなる配列は、このオブジェクトグループを配置するノード群とその状態（動作中，Join中等）を示している。尚更新伝播送信キューと実反映遅延キューについては後述する。

【００４６】

また状態フラグは、オブジェクトグループに属するファイルへのアクセス可否や、等価性回復中、系再構成中等の状態を表示するフラグの集合で、図４に示した各構成要素はこの状態フラグの対応ビットの１／０を切換えることよりこれらの状態を表示して他の構成要素に通知する。尚初期状態では、既に他のノードが系を作り、ファイルが更新されている可能性があるのでオブジェクトグループに

属するファイルは全てアクセス不可の状態として設定される。

【００４７】

初期処理が完了すると、系構成管理部１１はオペレータからオブジェクトグループに対する操作指令が投入されるのを待つ。

１）Joinコマンド投入

オペレータはオブジェクトグループに対する活性化を指示する場合、Joinコマンドを投入する。

【００４８】

このJoinコマンドが投入されると、系構成管理部１１は、他のノードとメッセージをやり取りしてJoinコマンドと共に指定されたオブジェクトグループに対する系に加わる。またJoinコマンドに単独での系生成を許可することを示すsingle指定がされていた場合、もしこのオブジェクトグループに対して系が構成されていなければ新たな系を生成する。

【００４９】

図７は、Joinコマンド投入時の系構成管理部１１による処理を示すフローチャートである。

Joinコマンドが投入されると、系構成管理部１１は、まずJoinコマンドと共に指定されたオブジェクトグループを共用している他のノードに順にメッセージを送り（ステップＳ１１）、返答を各ノードから受信する（ステップＳ１２）。

【００５０】

各ノードからの返答から、対象としているオブジェクトグループに対して既に系を作っているものでないかどうかを調べ、その結果既に系を作成しているという返答が他ノードからあれば（ステップＳ１３、ＹＥＳ）、そのノードにＪＯＩＮ要求を送り既存系への参入処理を依頼する（ステップＳ１４）。

【００５１】

この参入依頼に対し、ノードから参入失敗を通知する応答がされた場合（ステップＳ１５、ＹＥＳ）、Join失敗をオペレータに通知し（ステップＳ１６）、処理を終了する。また参入失敗の通知がなければ（ステップＳ１５、ＮＯ）、後述の参入処理（ステップＳ１７）を行った後、オペレータに成功応答を返す（ステ

ップＳ１８）。

【００５２】

またステップＳ１２の各ノードからの応答から、そのオブジェクトグループに対して未だ系を作っているノードがいないと判断され（ステップＳ１３、ＮＯ）、かつJoinコマンドのオプションでsingleが指定されていた場合（ステップＳ１９、ＹＥＳ）、このノードは自身のみで系を作る。

【００５３】

この際、系構成管理部１１は、まず、系状態テーブル内の情報を調べる。その結果、系状態テーブル中の整然停止フラグに整然停止が表示され自身が認識している最終の系状態が整然停止であると判断される時（ステップＳ２０、ＹＥＳ）、一定時間受信待機し（ステップＳ２１）、前回整然停止した時に共に系を構成していた他のノードが新規系への参入を依頼してくるのを待つ。そしてＪＯＩＮ要求により系への参入を依頼してきたノードに対し、順次後述する図９のＪＯＩＮ要求受け付け処理を行い、自己の系のバージョン番号を送信する。

【００５４】

この結果、全てのノードからＲＥＡＤＹ要求が到着したら（ステップＳ２２、ＹＥＳ）、ＲＥＡＤＹ要求に対する応答として、ＣＯＭＰＬＥＴＥ応答を全ノードに返す（ステップＳ２３）。また全てのノードからＲＥＡＤＹ要求が到着しなければ（ステップＳ２２、ＮＯ）、ＲＥＡＤＹ要求に対する応答としてＣＯＮＴ応答をノードに返し（ステップＳ２４）、更にＲＥＡＤＹ要求の到着を待つ。

【００５５】

ステップＳ２３でＲＥＡＤＹ要求に対する応答を送信した後、あるいはステップＳ２０で系状態テーブル内の整然停止フラグが前回の停止が整然停止でないことを示していた場合（ステップＳ２０、ＮＯ）、環境定義・状態情報２０の対応する系状態テーブルに記録されている系のバージョン番号をインクリメント（＋１）して更新する（ステップＳ２５）。そして、内部制御表の状態フラグをアクセス可能表示に変更して（ステップＳ２６）、ＩＯ要求インタセプト部１２に対応するオブジェクトグループへのアクセスが可能となったことを知らせる。そしてJoinコマンドに対する応答として処理完了をオペレータに通知して（ステップ

Ｓ２７）処理を終了する。

【００５６】

またステップＳ１９で、Joinコマンドのオプションとしてsingle指定がされていなかった場合には（ステップＳ１９、ＮＯ）、Joinコマンドに対する応答としてオペレータにエラーを通知し（ステップＳ２８）、処理を終了する。

２）参入処理

図８は、図７のステップＳ１７の系構成管理部１１の動作を示すフローチャートである。

【００５７】

ＪＯＩＮ要求による参入依頼に対し、参入失敗でなければ依頼先のノードから応答として系のバージョン番号が送信されてくる。この時系構成管理部１１は、まず現在系を構成するノード情報から内部制御表の依頼元ノードに対応するstatusをJoin中表示に更新し（ステップＳ３１）、次に応答で通知された既存系が保持しているバージョン番号と参入しようとしている自ノードが保持しているバージョン番号を比較する（ステップＳ３２）。その結果、２つのバージョン番号が異なる場合には、自ノードが系から脱落している間にオブジェクトグループ内のファイルに対し変更が加えられた可能性があることを示しているので、整然停止表示をリセットし（ステップＳ４１）、等価性回復処理を起動する（ステップＳ４２）。また２つの系のバージョン番号が一致していても系状態テーブルの整然停止フラグが非整然停止を表示していた場合には（ステップＳ３２、一致：ステップＳ３３、ＮＯ）、自己のファイルは最新のデータのものでないので、やはりステップＳ４２の等価性回復処理を起動する。そしてステップＳ４２の等価性回復処理の起動後は、処理の終了を待たずに応答値として送信されてきた系のバージョン番号を系状態テーブルに設定し（ステップＳ４３）た後、内部制御表の状態フラグをオブジェクトグループに対するアクセスが可能の表示に変更し（ステップＳ４０）、処理を終了する。

【００５８】

また送信されてきた系のバージョン番号と系状態テーブル内に記憶されている系のバージョン番号が一致しており（ステップＳ３２、一致）、かつ系状態テー

ブルの整然停止フラグが整然停止を表示しているなら（ステップＳ３３、ＹＥＳ）、自ノードが保持しているオブジェクトグループのファイルは最新のデータものなのでファイルの更新は必要ない。よって後述するステップＳ４２の等価性回復処理は行われず、ステップＳ３４として系のバージョン番号を更新後、定期的にＲＥＡＤＹ要求を送り（ステップＳ３５）、全ノードの参入が完了するのを待合わせる。

【００５９】

その結果ＲＥＡＤＹ要求に対する応答がＣＯＮＴ応答であれば（ステップＳ３６、ＣＯＮＴ）、一定時間後にＲＥＡＤＹ要求を再送し（ステップＳ３７）、同じ処理を繰り返す。またＲＥＡＤＹ要求の応答がＣＯＭＰＬＥＴＥ応答であればステップＳ３６、ＣＯＭＰＬＥＴＥ）、前回整然停止した時のノードが全て依頼元にＲＥＡＤＹ要求を行ったことになるので、応答で返される系を構成しているアクティブノードについての情報から、内部制御表のノード定義部の各ノードのstatusを動作中表示に変更する（ステップＳ３８）。

【００６０】

この後、系状態テーブルの整然停止表示をリセットし（ステップＳ３９）、内部制御表の状態フラグをオブジェクトグループがアクセス可能表示に変更し（ステップＳ４０）、処理を終了する。

３）ＪＯＩＮ要求受付処理

図９は、ＪＯＩＮ要求受付処理時の系構成管理部１１の動作を示すフローチャートである。

【００６１】

このＪＯＩＮ要求受付処理は、図７のステップＳ１４の新規参入依頼時に発行されたＪＯＩＮ要求や、ステップＳ２１の受信待機時に受け付けたＪＯＩＮ要求に対する処理を示したものである。

【００６２】

ＪＯＩＮ要求を行ったノードから受け付けたノードの系構成管理部１１は、ＪＯＩＮ要求と共に通知された依頼ノードの系のバージョン番号と系状態テーブル内の自身の系のバージョン番号とを比較する（ステップＳ５１）。その結果両方

のバージョン番号が一致しており（ステップＳ５１、一致）、また整然停止フラグを参照して整然停止後の整然立ち上げ中なら（ステップＳ５２、ＹＥＳ）、ステップＳ５３としてＪＯＩＮ要求に対する応答して自己の現在のバージョン番号を返答して処理を終了する。

【００６３】

ＪＯＩＮ要求と共に通知された情報から、２つの系のバージョン番号が一致しなかったり（ステップＳ５１、不一致）、一致しても整然停止後の系への参加でないのならば（ステップＳ５２、ＮＯ）、次に内部制御表のノード定義部を参照し、既にJoin中のノードが存在するかどうかを調べる（ステップＳ５４）。その結果、既にJoin中のノードが存在していれば、応答として失敗を通知して（ステップＳ５９）、処理を終了する。またJoin中のノードが他に存在しなければ（ステップＳ５４、ＮＯ）、このＪＯＩＮ要求により参入してきたノードに対応する内部制御表内のstatusを稼動中（アクティブ）、ＪＯＩＮ中（新規参入処理中）の表示に設定し（ステップＳ５５）た後、他のアクティブな全ノードにJoin通知を送る（ステップＳ５６）。そしてこのJoin通知に対する応答が全て返ってきた後に（ステップＳ５７、ＹＥＳ）、系のバージョン番号を更新し（ステップＳ５８）、ＪＯＩＮ要求に対する応答として現在の系のバージョン番号を返答して（ステップＳ５３）、処理を終了する。

４）Join通知

図１０は、図９のステップＳ５６で送信されたJoin通知を受取ったアクティブなノードの系構成管理部１１が行う処理を示すフローチャートである。

【００６４】

Join通知を受信すると、系構成管理部１１は、ステップＳ６１として内部制御表の、Join通知により通知された参入依頼をしているノードに対応するstatusを稼動中、Join中表示に設定する。そしてステップＳ６２として、Join通知に対する応答後、系状態テーブル内の系のバージョン番号を更新して（ステップＳ６３）処理を終了する。

５）等価性回復処理

図１１は、図８のステップＳ４２で起動される等価性回復処理の系構成管理部

１１の動作を示すフローチャートである。

【００６５】

等価性回復処理は、新規参入ノードが系から離脱している際に古くなった自己のファイル内のデータを復元する為の処理である。

等価性回復処理が起動されると、まず系構成管理部１１は内部制御表のノード定義部を参照して、系内のアクティブなノードの１つからオブジェクトグループ内の全ファイルのファイル名を取得する（ステップＳ７１）。

【００６６】

次にステップＳ７２として内部制御表の状態フラグを等価性回復中表示に設定した後、ステップＳ７３系内のアクティブなノードにステップＳ７１で得たファイル名を指定して転送要求を行う。この転送を等価性回復転送と呼ぶ。

【００６７】

このファイル転送に対する応答がエラーであったならば、転送要求先を他のアクティブなノードに変更して再度ファイル転送要求を行う（ステップＳ７５）。

ファイル転送要求に対して、要求先のノードから、正常応答を得たら（ステップＳ７４、正常）、ステップＳ７５として転送ファイルを受信し、これを受信データ処理部１５に自身のファイルにデータの反映を依頼する（ステップＳ７７）。この時通常のファイル更新に伴う更新データの伝播と、等価性回復処理での転送データの順序性は変更データ通知部１４及び受信データ処理部１５を介して保証されるので、等価性回復処理中にファイルを更新しても更新結果が失われることはない。

【００６８】

転送ファイルの受信及び自ファイルへの反映をステップＳ７１で得た全てのファイルに対して行い（ステップＳ７８、ＮＯ）、全ファイルへの処理が完了したならば（ステップＳ７８、ＹＥＳ）、ステップＳ７９として全アクティブノードに等価性回復処理の完了を通知し、全アクティブノードからの応答を待った後（ステップＳ８０）、内部制御表上の等価性回復処理中をリセットし（ステップＳ８１）、処理を終了する。尚ステップＳ７３～７８の等価性回復転送によるファイル転送は１つのノードに全ファイルの転送を要求してもよいし、複数のノード

に分散して要求してもよい。

６）等価性回復転送

図１２は、等価性回復処理を行っているノードから、図１１のステップＳ７３で送信される等価性回復転送要求を受信したノードの系構成管理部１１が行う処理を示すフローチャートである。

【００６９】

等価性回復転送を要求されたノードは、ステップＳ９１としてまずトークン管理ノードにWrite トークンの獲得を要求する。その結果、Write トークンを獲得できなければ（ステップＳ９２、ＮＯ）、要求先のノードにエラー応答をして（ステップＳ９３）処理を終了する。

【００７０】

またWrite トークンを獲得できれば（ステップＳ９２、ＹＥＳ）、ステップＳ９３として要求元のノードに正常を応答した後、ステップＳ９５として要求されたファイルデータを変更データ通知部１４を介して順次要求元のノードに転送し、処理を終了する。

７）等価性回復完了メッセージ

図１３は、等価性回復処理が完了した最新のデータにファイルの復元がなされたノードが、図１１のステップＳ７９で送信した等価性回復完了メッセージを受信した系内のアクティブなノードが行う処理を示すフローチャートである。

【００７１】

等価性回復完了メッセージを受信したノードは、ステップＳ９６として内部制御表内の送信もとノードに対応するstatusに表示されているＪＯＩＮ中の表示をリセット後、ステップＳ９７としてメッセージの送信元ノードに応答を返して処理を終了する。

【００７２】

この図１３の処理により、新規参入してきたノードは系内の他のアクティブノードから系への参入処理が完了したものとみなされる。

８）Join再試行メッセージ

Join中に系の再構成が発生すると、このJoin再試行メッセージがJoin中のノー

ドに送られる。この要求を受けた系構成管理部１１は、系への新規参入処理を最初からやり直す。

９）停止処理

ノードを停止させる場合、オペレータは系からの離脱を指示するleave コマンドを投入して系から離脱する。尚ここでの停止とは、ノードが系から離脱することを示しており、ノードが複数の系に属している場合、メンテナンス等でノードを完全に止めるためには各系に対してleave コマンドを投入して全ての系から離脱しなければならない。

【００７３】

ノードの停止をleave コマンドでオペレータから通知されると、系構成管理部１１は以下の処理を行う。

ａ）整然停止

整然停止は、系を構成している全ノードが同期して一斉に停止し系そのものが停止するもので、正月休みやシステムの再構築等の場合にシステム全体を休止させるために行われる。オペレータは整然停止を行う場合、オプションでall を指定したleave コマンドを投入する。

ｂ）非整然停止

非整然停止は、そのノードのみを停止させるものであり、非整然停止したノードのみ系から離脱し、他のノードによって系は存続する。オペレータは非整然停止を行う場合、オプションでall を指定しないでleave コマンドを投入する。

【００７４】

図１４は、オペレータがleave コマンドを投入して、ノードの停止を指示した時の系構成管理部１１の処理を示すフローチャートである。

leave コマンドが投入されると、系構成管理部１１は、まずステップＳ１０１として内部制御表の状態フラグをアクセス不可表示に変更し、図４の他の構成要素に（具体的にはＩＯ要求インタセプト部１２に）対応するオブジェクトグループに属するファイルへのアクセスを禁止する。

【００７５】

次に系構成管理部１１は、ステップＳ１０２として変更データ通知部１４にＳ

ＹＮＣ要求を行い、キューに保持され遅延している更新要求の全ノードへの反映を依頼する。

【００７６】

全ノードへの変更データの反映が完了し、変更データ通知部１４から完了が通知されると（ステップＳ１０３、ＹＥＳ）、leave コマンドにall 指定が無い場合には（ステップＳ１０４、ＮＯ）、非整然停止なので処理を終了する。

【００７７】

またステップＳ１０４でall 指定がある場合には、整然停止を行うため、ステップＳ１０５として整然停止開始メッセージを系内の全ノードへ一定時間送信し、整然停止開始メッセージに対する応答が全ノードから返信されるのを待つ（ステップＳ１０６）。そして全ノードから応答があると（ステップＳ１０６、ＮＯ）、整然停止を行ったオブジェクトグループに対応する系状態テーブル内の整然停止フラグを整然停止にセットして（ステップＳ１０７）、処理を終了する。

１０）ノード障害認識

障害等による他のノードの離脱は、例えば分散システムで一般的に行われている自己の存在を他ノードに通知するメッセージ（I'm alive メッセージ）を送信し合うグループコミュニケーションシステムにおいて、メッセージが途絶えたり、応答が返ってこない等の場合に、系内の他のノードによって認識される。系内の他ノードの離脱を認識したノードは、系の再構成を系内の他のアクティブなノードに要求する。

【００７８】

図１５は、系内の他ノードの離脱を認識したノードの系構成管理部１１の処理を示すフローチャートである。

現在系を構成しているノードの障害を認識すると、系構成管理部１１はまずステップＳ１１１として、内部制御表の状態フラグを系再構成中を表示するよう設定し、変更データ通知部１４にメッセージを他のノードに送るのを一時抑止させる。

【００７９】

次に系構成管理部１１は、ステップＳ１１２として、系の再構成要求メッセー

ジを系内の全アクティブノードに送信して他のノードの系構成管理部１１とやり取りし、系の再構成の合意を得る。この時、もしJoin中のノードを除く過半数のノードから合意が取れなければ（ステップＳ１１３、ＮＯ）、状態フラグをアクセス禁止の表示にセットして（ステップＳ１１４）、対応するオブジェクトグループ内のファイルへのアクセスを禁止した後、ステップＳ１１１でセットした系再構成中の表示をリセットして（ステップＳ１１５）、処理を終了する。

【００８０】

またステップＳ１１３で、系の再構成要求に対してJoin中のノードを除く過半数のノードから合意が取れると（ステップＳ１１３、ＹＥＳ）、系状態テーブル内の系のバージョン番号を更新し（ステップＳ１１６）、ノード定義部の各ノードのstatusを変更して合意の取れた過半数のノードを新しいアクティブなノードとして内部制御表に設定して（ステップＳ１１７）、最新の系状態を表すように更新する。

【００８１】

この後、変更データ通知部１４にＲＥＳＥＴ要求を送り（ステップＳ１１８）、応答を待つ（ステップＳ１１９）。変更データ通知部１４から応答があったら（ステップＳ１１９、ＹＥＳ）、更新伝播送信キュー内の変更内容の他ノードへの伝播完了を通知するＲＥＳＥＴＣＯＭＰをアクティブな全ノードの系構成管理部１１に送り、全ノードからＲＥＳＥＴＣＯＭＰに送られてくるのを待つ（ステップＳ１２１）。

【００８２】

全ノードからＲＥＳＥＴＣＯＭＰが送られてきたら（ステップＳ１２１、ＹＥＳ）、伝播中であったファイルの更新要求が全て自ノードに到着したことになるので、ステップＳ１２２として受信データ処理部１５にＲＥＳＥＴ要求を送り、系から切り離されたノードに関する送信、受信の後始末を依頼し処理完了通知を待つ（ステップＳ１２３）。

【００８３】

受信データ処理部１５から処理の完了が通知されると（ステップＳ１２３、Ｙ）、ステップＳ１１１でセットした系再構成中の表示をリセットして（ステップ

Ｓ１２４）、処理を終了し、通常処理を再開させる。

【００８４】

尚Join中のノードには、Join再試行要求を送り、最初から系への新規参入処理をやり直させる。

［ＩＯ要求インタセプト部］

ＩＯ要求インタセプト部１２は、ユーザプログラム１７が発行したファイルへのアクセス要求を受取り、ＯＳ内のファイルシステムにこのアクセス要求を伝える部分で、ユーザプログラム１７がファイルに対する入出力要求を発行すると、ＩＯ要求インタセプト部１２に制御が渡る。

【００８５】

ＩＯ要求インタセプト部１２は要求されたファイルの名前が全ての内部制御表に設定されているいずれのパス内にも属していないなら、直ちにＯＳのファイルシステムに制御を渡す。そしてファイルシステムから戻された応答をユーザプログラム１７に返す。

【００８６】

またもしそのファイルが、複数の内部制御表の内のオブジェクトグループ定義部内に定義されているいずれかのパスに属するものであるならば、要求されたファイルへのアクセス要求がオブジェクトグループに属するものと見なし、以下の処理を行う。

１）アクセス不可表示が内部制御表にある場合

オブジェクトグループへのアクセスは禁止されているので、ユーザプログラム１７にエラーを応答する。

２）等価性回復中の場合

稼動中の他ノードにＦＯＲＣＥ指定のRead要求若しくはWrite 要求を送り、ファイルへのアクセスを依頼する。系内の他のノード（Join中を除く）は、最新データのファイルを保持しているので、Read/Write要求に対して応答データを送信してきた場合には、このデータは一貫性が保証されているものなのでこれをユーザプログラム１７に返す。またRead/Write要求に対して失敗を応答されたら、別の稼動中ノードに対して同様の処理を繰り返す。

３）等価性回復中でない場合

ａ）Write 系要求

要求されたファイルのWrite トークン獲得をトークン管理部１３に依頼する。トークン管理部１３から獲得成功を応答された場合、ＯＳのファイルシステムを呼び、自身のファイルに対しデータの更新処理行った後、変更内容を変更データ通知部１４に渡して他ノードへの反映を行う。

【００８７】

トークン管理部１３からWrite トークン獲得失敗を応答されたら、トークン管理ノードから応答時に通知されたWrite トークン保持ノードにWrite 要求を送り処理を依頼する。またWrite トークン保持ノードからWrite 要求に対して処理失敗（トークン変化）を応答されたら、トークン獲得からやり直す。

【００８８】

尚自ノードのファイルを更新する際の待合わせ処理や、他ノードの受信データ処理部１５に送るWrite 要求に付加するデータなど、ＩＯ要求インタセプト部１２で行われる順序性保証処理は後述する。

ｂ）Read系要求

要求されたファイルのReadトークン獲得をトークン管理部１３に依頼する。トークン管理部１３から獲得成功を通知されたら、ＯＳのファイルシステムを介し自ノードのファイルからデータを読み、ユーザプログラム１７に応答する。

【００８９】

トークン管理部１３からReadトークン獲得失敗を応答されたら、応答時に通知されたノード（Write トークン保持ノード）にRead要求を送る。成功応答が要求先のノードからあれば、渡されたデータをユーザプログラム１７に返す。また失敗（トークン変化）を応答されたらトークン獲得からやり直す。

【００９０】

尚他ノードで行われた更新の待合わせなど、順序性保証に伴う処理は後述する。

尚Read/Writeトークンの獲得／解放はユーザプログラム１７からのRead/Write要求発行単位で行う構成にする他、オーバヘッドを減らすためファイルのOpen/C

lose単位に行う構成にしても良い。この場合、ユーザプログラムがファイルをオープンした時点で上記トークン処理が行われ、クローズが発行されるまでトークンが保持される。またトーク獲得不可をオープン時に通知された場合、以降のＩＯ要求はWrite トークンを保持しているノードに転送される。

【００９１】

また、トークン解放を自発的に行うのではなく、ファイル処理が完了するとトークンを必要としていないことを表示しておき、他ノードがトークンを必要とするタイミングまで解放を遅らせる構成とすることも出来る。尚、Write 時及びRead時には後述する順序性保証処理も行われる。

【００９２】

図１６は、ＩＯ要求インタセプト部による処理を示すフローチャートである。

ユーザプログラム１７からファイルへのアクセス要求が発行されるとＩＯ要求インタセプト部１２はまず内部制御表を参照し、要求されたファイルのファイル名とオブジェクトグループ定義部内のパス名を比較する（ステップＳ１３１）。その結果一致しなければ（ステップＳ１３１、不一致）、要求されたファイルはオブジェクトグループに属していないので、ＯＳのファイルシステムの制御を渡し（ステップＳ１３２）、ファイルへの処理を依頼する。そしてファイルシステムからの応答をユーザプログラムに返して（ステップＳ１３３）、処理を終了する。

【００９３】

ステップＳ１３１でファイル名が内部制御表内のいずれかのパスに属するものであるのならば（ステップＳ１３１、一致）、そのファイルはオブジェクトグループに属するものであるので対応する内部制御表の状態フラグを調べる。その結果アクセス不可が表示されていれば（ステップＳ１３５、ＹＥＳ）、ステップＳ１３４としてユーザプログラム１７にエラー応答を行い処理を終了する。

【００９４】

また状態フラグに等価性回復中表示がされていた場合には（ステップＳ１３６、ＹＥＳ）、ステップＳ１５０として稼動中の他ノードにオプションでＦＯＲＣＥ指定をしたRead/Write要求を送り、応答を待つ（ステップＳ１５１）。その結

果失敗を応答されたら（ステップＳ１５２、失敗）、ステップＳ１５３として別のアクティブなノードにＦＯＲＣＥ指定のRead/Write要求を送り、応答を待つ。またRead/Write要求を送ったノードから成功応答があると（ステップＳ１５２、成功）、ステップＳ１５４として応答データをユーザプログラム１７に応答して処理を終了する。

【００９５】

状態フラグに、アクセス不可と等価性回復中のいずれもが表示されていない時（ステップＳ１３５及び１３６、ＮＯ）、ＩＯ要求インタセプト部１２は、アクセス要求がRead要求であった時（ステップＳ１３７、Ｒｅａｄ）、ステップＳ１４４として要求されたファイルのReadトークン獲得をトークン管理部１３に依頼する。

【００９６】

その結果トークン管理部１３からトークンの獲得成功を通知されたら（ステップＳ１４５、ＹＥＳ）、ＯＳファイルシステムを介し自ノードのファイルからデータを読みだし（ステップＳ１４６）、これをユーザプログラムへ返答（ステップＳ１４７）後、トークンを自発的に解放する構成の場合トークン管理部１３にトークンを解放を依頼してから処理を終了する。またトークン管理部１３からReadトークンの獲得失敗を応答された場合には（ステップＳ１４５、ＮＯ）、ステップＳ１４８として失敗と共に通知されたWrite トークン保持ノードにRead要求を送信し、応答を待つ。その結果Write トークン保持ノードから、成功を通知された場合には（ステップＳ１４９、成功）、渡されたデータをユーザプログラム１７に応答後（ステップＳ１４７）、処理を終了する。またWrite トークン保持ノードから、失敗を通知された場合には（ステップＳ１４９、失敗）、ステップＳ１４４のReadトークン獲得依頼から処理をやり直す。尚ステップＳ１４６のデータ読みだし時に読み出し対象となっているファイルの転送モードが非同期若しくは半同期モードである場合、実反映遅延キューを参照して最新のデータがキューイングされていればそちらを読み出す。この点については順序性保証の項で詳細に説明する。

【００９７】

またステップＳ１３７でアクセス要求がWrite 要求であった場合には（ステップＳ１３７、Ｗｒｉｔｅ）、ステップＳ１３８として要求されたファイルのWrite トークン獲得をトークン管理部１３に依頼する。

【００９８】

その結果トークン管理部１３からトークン獲得成功を応答されたならば（ステップＳ１３９、ＹＥＳ）、ステップＳ１４０としてＯＳのファイルシステムを呼び自身のファイルに対するWrite 処理を依頼し、ステップＳ１４１としてデータの変更内容を変更データ通知部１４に渡して他ノードへの反映を依頼した後、トークン解放を自発的に行う構成の場合トークン管理部１３にトークンの解放を依頼して処理を終了する。またトークン管理部１３からWrite トークン獲得失敗を応答された場合には（ステップＳ１３９、ＮＯ）、ステップＳ１４２として応答時に通知されたWrite トークン保持ノードにWrite 要求を送り、応答を待つ。その結果失敗を応答されたならば（ステップＳ１４３、失敗）、ステップＳ１３８のWrite トークン獲得からやり直す。また成功を応答されたならば（ステップＳ１４３、成功）、後述する順序性保証処理による更新内容の自ファイルへの反映を考慮しつつ処理を終了する。尚ステップＳ１４０で、ファイルに対する処理を依頼する際、対象ファイルの伝播モードが非同期若しくは半同期モードである場合、変更内容は実反映遅延キューにキューイングして順序性保証を考慮した処理が行われる。この点については順序性保証の項で詳細に説明する。

［トークン管理部］

トークン管理部１３は、ファイルアクセス権限を管理する部分で、系を構成する全ノードが同じ情報を保持するように制御を行う。尚実装を簡単にする為、系を構成するいずれか１つのノード（例えばネットワークアドレスが一番小さいノード）をトークン管理ノードとし、トークン管理ノードのトークン管理部１３をサーバとして系全体の全トークン状態を保持、管理する構成とし、他のノードのトークン管理部１３は、クライアントとして自ノードが保持しているトークンのみを管理する構成とするのが一般的である。

【００９９】

トークン管理ノードのトークン管理部１３は、メモリ上にトークン制御表を構

成し、このトークン制御表により系内に存在する全ノードを管理する。

図１７は、トークン制御表の構成例を示す図である。

【０１００】

同図では、トークン制御表はリストデータ構造を取っており、各オブジェクトグループに属するファイル毎に１つ対応するトークン制御表が生成される。トークン制御表はトークンがオブジェクトグループ内のどのファイルに対してのものであるかを示すファイル識別子、トークンの種類（Read/Write）を示すトークン状態、トークンを保持しているノードを指定する保持ノード番号及び次の制御表の一を示すポインタが記憶されている。このうちトークン識別子にはトークン管理部１３が対応する制御表を検索するためのタグとなるもので、対応するファイルのファイル名等が用いられる。リストの検索を速くするためにファイル識別子にハッシュ関数を適用し、得られた値が同じものが一つのキューを構成するように構成される。

【０１０１】

トークン管理ノードのトークン管理部１３は、自ノードのＩＯ要求インタセプト部１２や他ノードのトークン管理部１３からトークンに対する処理要求があるとこのトークン制御表を検索し、要求されたファイルのトークンの状況を調べる。またトークンを生成したり解放する時は、新たなトークン制御表をリストデータに加えたり、対応するトークン制御表をリストデータから削除する。

【０１０２】

また系を再構成した場合には、各ノードが保持する最終のトークン保持情報から系全体のトークン状態を復元する。

図１８はトークン管理ノードのトークン管理部１３の処理を示すフローチャートである。

【０１０３】

トークン管理部１３は、他ノードのトークン管理部１３や自ノードのＩＯ要求インタセプト部１２からトークンに対する処理要求を受取ると、以下の様に処理する。

【０１０４】

トークン管理部１３は、他ノードのトークン管理部１３や自ノードのＩＯ要求インタセプト部１２から処理要求を受取とるとまず、要求内容を判断する（ステップＳ１５１）。その結果、Write トークン獲得要求であるのならば、ステップＳ１５２として図１９のWrite トークン獲得要求処理を行い、Readトークン獲得要求であるのならば、ステップＳ１５３として図２０のReadトークン獲得要求処理を行い、トークン解放要求若しくはトークン回収要求であるのならば、ステップＳ１５４として、トークン解放／回収要求処理を行ったのち、処理を終了する。

【０１０５】

図１９は、図１８のステップＳ１５２のWrite トークン獲得要求処理時のトークン管理部１３の処理動作を示すフローチャートである。

トークン管理部１３は、Write トークン獲得要求処理ではまずトークン制御表を参照して、Write トークン獲得要求を行っているノードがWrite トークンを保持しているかどうかを調べる（ステップＳ１６１）。その結果、保持していた場合（ステップＳ１６１、ＹＥＳ）、トークン獲得成功を要求元ノードに応答し（ステップＳ１６８）、処理を終了する。また要求元のノードがWrite トークンを保持していない場合（ステップＳ１６１、ＮＯ）、次に要求元以外のノードが要求されているファイルへのWrite トークンを保持しているかどうかを判断する。その結果Write トークンを保持しているノードがある場合（ステップＳ１６２、ＹＥＳ）、Write トークン獲得不可をWrite トークンを保持しているノードのノード番号と共に応答し（ステップＳ１６３）、処理を終了する。

【０１０６】

またWrite トークンを保持しているノードが存在しない場合には（ステップＳ１６２、ＮＯ）、他ノードが要求されているファイルのReadトークンを保持しているか判断する。その結果Readトークンを保持しているノードが存在しなければ（ステップＳ１６４、ＮＯ）、トークン制御表を操作して要求元ノードにWrite トークンを渡し（ステップＳ１６８）、トークン獲得成功を要求元ノードに応答して処理を終了する。またReadトークンを保持しているノードが存在すれば（ステップＳ１６５、ＹＥＳ）、ステップＳ１６６としてReadトークンを保持してる

全てのノードにトークン回収を指示し、全Readトークン保持ノードから回収完了を通知されるのを待ち（ステップＳ１６６、ＮＯ）、全てのReadトークンの回収が完了した後（ステップＳ１６６、ＹＥＳ）、要求元ノードにWrite トークンを渡し（ステップＳ１６７）、トークン獲得成功を要求元ノードに応答して（ステップＳ１６８）処理を終了する。

【０１０７】

図２０は、図１８のステップＳ１５３のReadトークン獲得要求処理時のトークン管理部１３の処理動作を示すフローチャートである。

Readトークン獲得要求処理では、トークン管理部１３はまずトークン制御表を参照して、Readトークン獲得要求を行っているノードが、Readトークン若しくはWrite トークンを保持しているかどうか調べる（ステップＳ１７１）。その結果どちらかのトークンを保持していた場合（ステップＳ１７１、ＹＥＳ）、トークン獲得成功を要求元ノードに応答し（ステップＳ１７５）、処理を終了する。また要求元のノードがReadトークン、Write トークン共に保持していない場合（ステップＳ１７１、ＮＯ）、次に要求元以外のノードが要求されているファイルへのWrite トークンを保持しているかどうかを判断する。その結果Write トークンを保持しているノードがある場合（ステップＳ１７２、ＹＥＳ）、Readトークン獲得不可をWrite トークンを保持しているノードのノード番号と共に応答し（ステップＳ１７３）、処理を終了する。

【０１０８】

またWriteトークンを保持しているノードが存在しない場合には（ステップＳ１７２、ＮＯ）、トークン制御表を操作して要求元ノードにReadトークンを渡し（ステップＳ１７３）、ステップＳ１７４としてトークン獲得成功を要求元ノードに応答して処理を終了する。

【０１０９】

図２１は、図１８のステップＳ１５４トークン解放／回収要求処理時のトークン管理部１３の処理動作を示すフローチャートである。

トークン解放要求は、トークンが不要となったノードが行うもので、系内の全ノードに更新データの伝播が完了した時等に発行される。尚不要になったトーク

ンの解放を自発的に行わない構成の場合、トークン解放要求を受けた時点ではトークン保持ノード側のトークン管理部１３はトークン返却可能を表示しておくのみで復帰する。この場合、Write トークン獲得要求処理やReadトークン獲得要求処理では、トークン管理ノードのトークン管理部１３はWrite トークン保持ノードに対してもトークン回収を指示する。そして、トークンを保持しているノードからトークン回収完了を通知された場合にはトークンを獲得できたとして処理を行い、回収不可を通知された場合にWrite トークンを保持しているノードが存在すると見なして処理を行う。

【０１１０】

またトークン回収要求は、Write トークン獲得要求処理時にトークン管理ノードのトークン管理部１３がRead/Writeトークンを保持しているノードに対して発行する要求である。Write トークンに対する回収要求は、トークン保持ノードが不要になったトークンを自発的に返却しない構成の場合にのみ発行される。

【０１１１】

トークン解放要求若しくはトークン回収要求を受けたトークン管理部１３は、ステップＳ１８１として指定されたトークンを直ちに解放し、解放に成功したことをトークン管理ノードのトークン管理部１３に応答して（ステップＳ１８２）処理を終了する。

【０１１２】

図２２は、不要になったトークンを自発的に返却しない構成の場合に発行されるWrite トークン回収要求を受けたWrite トークン保持ノードが行う処理を示すフローチャートである。

【０１１３】

Write トークン回収要求を受けると、Write トークンを解放できる状態にあるかどうか判断される（ステップＳ１９１）。その結果、該当ファイルへの書込み処理が完了しておらず、Write トークンを解放できない状態にある場合（ステップＳ１９１、ＮＯ）、Write トークン回収要求を送信してきたトークン管理ノードのトークン管理部１３にWrite トークン解放失敗を応答し（ステップＳ１９６）、処理を終了する。

【０１１４】

またWrite トークンを解放可能な状態である時は（ステップＳ１９１、ＹＥＳ）、まずステップＳ１９２としてＦＳＹＮＣ指定で変更データ通知部１４を呼び、更新伝播送信キューにキューイングされている自己が行ったファイルの変更や他ノードから依頼されているファイルの変更内容を全て、系内の全ノードへの伝播を依頼し、完了応答を待つ（ステップＳ１９３、ＮＯ）。

【０１１５】

そして全ノードから応答があり、変更データ通知部１４から伝播完了を通知されると（ステップＳ１９３、ＹＥＳ）、ステップＳ１９４としてWrite トークンを解放した後、トークン管理ノードのトークン管理部１３にトークン解放成功を応答して（ステップＳ１９５）、処理を終了する。

[変更データ通知部]

変更データ通知部１４は、ＩＯ要求インタセプト部１２または受信データ処理部１５からファイルの更新データを受取り、ファイルの変更内容の他ノードへの反映をスケジュールする部分である。

【０１１６】

変更データ通知部１４は、通知されたファイルが属するオブジェクトグループの系状態テーブルに設定されている伝播モード（同期，非同期，半同期）に従い、以下の様に処理する。

【０１１７】

尚、同期、半同期、非同期はユーザがオブジェクトグループ単位に信頼性要件に従って選択するものであるが、おおよそ以下の様な特性をもつ。

同期：ユーザプログラム１７が発行したファイルのへの書込み要求の完了がユーザプログラム１７に通知された時点で、ファイルへの更新データが他のノード全てに伝播されている保証が与えられる。従って、全ノードが壊れない限り、データが失われることはない。

【０１１８】

半同期：ユーザプログラム１７が発行したWrite 命令の完了がユーザプログラム１７に通知された時点で、更新結果が過半数のノードに伝播している保証が与

えられる。従って、半分以上のノードが同時に壊れない限り、データが失われることはない。すなわち、ノード障害に伴う系の縮退では過半数以上のノードで新しい系を作成するので、データが失われることはない。

【０１１９】

非同期：ユーザプログラム１７が発行したWrite 命令の完了がユーザプログラム１７に通知された時点で、更新結果が他のノードに伝播している保証はない。従って、ノード障害が発生すると、完了した筈の更新結果が失われることがある。但し本実施形態のシステムでは、この場合でも更新の順序性は保証されるので、新旧のデータが入り交じって見えることはない。

１）同期モード伝播時の処理

オブジェクトグループを構成するアクティブな全ノードに変更内容を転送し、全ノードから受信応答が戻ったところで、要求元に復帰する。

２）半同期モード伝播時の処理

オブジェクトグループを構成するアクティブな全ノードに変更内容を転送し、過半数のノードから受信応答が全て戻ったところで要求元に復帰する。尚Writeトークンは全てのノードへの伝播が完了するまでは解放しない。

３）非同期モード伝播時の処理

変更内容をメモリ上にターゲットノード単位でキューイングし、適当なタイミングで転送する。

【０１２０】

ここで、適当なタイミングとは以下のいずれかの状態が発生した時を指す。

１）系構成管理部１１からＳＹＮＣ要求を受け付けた時。すべての更新データを全ノードに伝播させる。

２）トークン管理部１３からWrite トークンを返却する前に、ＦＳＹＮＣ指定で呼ばれ、対象ファイルに対する変更内容を全ノードに伝播させる。

３）システムが判断した適当なタイミング。例えば一定時間立った時、あるいはキューイングされたデータが一定以上になった時。

【０１２１】

図２３は変更データ通知部１４による処理を示すフローチャートである。

変更データ通知部１４は、他の構成要素から呼び出されると、まず自己を呼び出した相手を判断する（ステップＳ２０１）。その結果、ＩＯ要求インタセプト部１２若しくは受信データ処理部１５により呼び出されたのであれば、ステップＳ２０２のＩＯ要求インタセプト部／受信データ処理部呼び出し処理を行う。また呼び出し元が系構成管理部１１であり、要求内容がＳＹＮＣ要求であるのならば（ステップＳ２０３、ＳＹＮＣ）、ステップＳ２０４のＳＹＮＣ要求処理を行い、また要求内容がＲＥＳＥＴ要求であるのならばステップＳ２０５のＲＥＳＥＴ要求処理を行う。またトークン管理部１３からＦＳＹＮＣ要求によって呼ばれた場合には、ステップＳ２０６のＦＳＹＮＣ要求処理を行って処理を終了する。

【０１２２】

図２４は、図２３のステップＳ２０２のＩＯ要求インタセプト部／受信データ処理部呼び出し処理の動作処理を示すフローチャートである。

ＩＯ要求インタセプト部／受信データ処理部呼び出し処理に入ると、変更データ通知部１４は、呼び出し元から通知された更新要求のオブジェクトグループ番号から対応するオブジェクトグループの内部制御表を見つけ、伝播モードを調べる（ステップＳ２１１）。　次にステップＳ２１２として更新要求を更新伝播キューの最後につないでキューイングする。そして、ステップＳ２１１で調べた伝播モードが非同期方式であったならば（ステップＳ２１３、非同期）、処理を終了し、呼び出し元に復帰する。

【０１２３】

伝播モードが同期方式か半同期方式の場合（ステップＳ２１３、同期／半同期）、内部制御表の状態フラグに系再構成中が表示されていた場合には、系の再構成が完了して状態フラグの表示が消えるのを待ち合わせた後（ステップＳ２１４）、ステップＳ２１５として系内の全アクティブノードに更新要求を送信する。

【０１２４】

更新要求送信後、更新伝播送信キュー内のack 待ちベクタの更新要求を送信したノードに対応するビットを立て（ステップＳ２１６）、応答を待つ。そして伝播モードが半同期の場合には（ステップＳ２１７、半同期）、ack ベクタの過半数がオフになり、更新要求を送信したノードの受信データ処理部１５の過半数か

ら受信完了の応答があるまで待ち合わせ（ステップＳ２１８）、処理を終了して、要求元に復帰する。

【０１２５】

また伝播モードが同期であった場合には（ステップＳ２１７、同期）、ステップＳ２１９としてack 待ちベクタが全てオフになるのを待合わせ、トークンを自発的に返却する構成の場合トークンを解放した後に処理を終了し、要求元に復帰する。

【０１２６】

図２５は、図２３のステップＳ２０４のＳＹＮＣ要求処理時の変更データ通知部１４の動作処理を示すフローチャートである。このＳＹＮＣ要求処理は、更新伝播送信キュー内にキューイングされている変更要求を全て系内の他ノードに伝播させて更新伝播送信キューにキューイングされている更新要求を全て掃き出させるもので、系構成管理部１１からＳＹＮＣ要求により呼ばれた時に行われる。

【０１２７】

ＳＹＮＣ要求処理に入ると、変更データ通知部１４は、まずステップＳ２２１として内部制御表内の更新伝播送信キューのエントリを用いて更新伝播送信キューの先頭要素を読み出す。

【０１２８】

図２６は、更新伝播送信キューの構成例を示す図である。

更新伝播送信キューは、更新要求をキューイングするバッファで、内部制御表内の更新伝播送信キューエントリによって先頭要素の位置が示されるリスト構造を持つ。リスト構造の１つの要素は１つの更新要求に対応しており、変更データ通知部１４は更新要求が生じると、更新伝播送信キューの最後に新規の要素を繋ぎ、処理が完了すると対応する要素を削除する。

【０１２９】

リストデータの１つの要素は、次の要素の位置を示すポインタ、更新を行うファイルが属するオブジェクトグループのオブジェクトグループ番号、この更新要求を他ノードに送信したかどうかを示す送信済みフラグ、各ノード毎の応答状態を示すack 待ちベクタ、更新対象ファイルのファイル名とそのファイル中での更

新位置をするオフセット、更新データの大きさを示す長さ、更新要求を行ったノードのノード番号を示す要求ノード番号、更新番号、依存ベクタ、更新内容を示す更新データによって構成される。これらのうち更新番号及び依存ベクタは後述する順序性保証処理で用いられるもので、順序性保証の項で詳細に説明する。

【０１３０】

ステップＳ２２１で読み出した要素の送信済みフラグが未送信を表示していたならば（Ｓ２２２、ＮＯ）、ステップＳ２２３としてこの要素の更新要求を系内の全アクティブノードに送信し、送信したノードに対応するack ベクタのビットを立てる（ステップＳ２２４）。また読み出した要素の送信済みフラグが送信済みを表示しており、この送信要求が他ノードに伝播中のものであったならば（ステップＳ２２２、ＹＥＳ）、その要素はスキップする。

【０１３１】

そして次の更新伝播送信キュー内の次の要素を読みだし（ステップＳ２２５、ＹＥＳ：ステップＳ２２６）、ステップＳ２２２～Ｓ２２４の処理を繰り返す。

キュー内の全要素に対して処理が完了すると（ステップＳ２２５、ＹＥＳ）、更新伝播送信キューの全要素のack 待ちベクタが０にリセットされ、更新要求を送った全てのノードから受信完了の応答があるのを待ってから（ステップＳ２２７）、処理を終了して要求元に復帰する。

【０１３２】

図２７は、図２３のステップＳ２０５のＲＥＳＥＴ要求処理時の変更データ通知部１４の動作処理を示すフローチャートである。ＲＥＳＥＴ要求は、障害発生時に伝播途中であった要求を全ノードに伝播させ、新しい系の同期を取るなどの目的で用いられる。このＲＥＳＥＴ要求処理は、他ノードのノード障害を認識した系構成管理部１１に、ＲＥＳＥＴ要求によって呼び出された変更データ通知部１４が行う処理である。ＲＥＳＥＴ要求処理では更新伝播送信キュー及び実反映遅延キューにキューイングされている更新要求を全て他ノードに伝播して更新内容を全他ノードに反映させる。

【０１３３】

ＲＥＳＥＴ要求処理に入ると、変更データ通知部１４は、ステップＳ２３１と

して図２６に示したＳＹＮＣ要求処理と同様の処理を行い更新伝播送信キューにキューイングされている変更要求を全て系内の他ノードに伝播して変更内容を通知する。

【０１３４】

次にステップＳ２３２として、内部制御表内の実反映遅延キューのエントリから位置を調べ、実反映遅延キューの先頭要素を読み出す。

ステップＳ２３２で読み出した要素の送信済みフラグが未送信を表示していたならば（Ｓ２３３、ＮＯ）、ステップＳ２３４としてこの要素の更新要求を系内の全アクティブノードに送信し、送信したノードに対応するack ベクタのビットを立てる（ステップＳ２３５）。また読み出した要素の送信済みフラグが送信済みを表示しており、この送信要求が他ノードに伝播中のものであったならば（ステップＳ２３３、ＹＥＳ）、その要素はステップＳ２３４及び２３５をスキップする。

【０１３５】

そして次の実反映遅延キュー内の次の要素を読みだし（ステップＳ２３６、ＮＯ：ステップＳ２３７）、ステップＳ２３３～Ｓ２３５の処理を繰り返す。

キュー内の全要素に対して処理が完了すると（ステップＳ２３６、ＹＥＳ）、実反映遅延キューの全要素のack 待ちベクタが０にリセットされ、更新要求を送った全てのノードから受信完了の応答があるのを待ってから（ステップＳ２３８）、処理を終了して要求元に復帰する。

図２８は、図２３のステップＳ２０６のＦＳＹＮＣ要求処理時の変更データ通知部１４の動作処理を示すフローチャートである。このＦＳＹＮＣ要求処理は、変更データ通知部１４が系構成管理部１１からファイル名を指定してＦＳＹＮＣ要求されて実行されるもので、Writeトークンを解放する目的などで、更新伝播送信キュー内にキューイングされている変更要求の内指定されたファイルに対するもの全てを系内の他ノードに伝播させて更新伝播送信キューから掃き出させるものである。

【０１３６】

ＦＳＹＮＣ要求処理に入ると、変更データ通知部１４は、まずステップＳ２４１として内部制御表内の更新伝播送信キューのエントリを用いて更新伝播送信キューの先頭要素を読み出す。

【０１３７】

ステップＳ２４１で読み出した要素内のファイル名と指定されたファイル名と比較し同一のものであり（ステップＳ２４２、ＹＥＳ）、また送信済みフラグが未送信を表示していたならば（Ｓ２４３、ＮＯ）、ステップＳ２４４としてこの要素の更新要求を系内の全アクティブノードに送信し、送信したノードに対応するack ベクタのビットを立てる（ステップＳ２４５）。また要素内のファイル名が指定されたものと異なったり（ステップＳ２４２、ＮＯ）、ファイル名は同じであっても読み出した要素の送信済みフラグが送信済みを表示しており、この送信要求が他ノードに伝播中のものであったならば（ステップＳ２４３、ＹＥＳ）、その要素はスキップする。

【０１３８】

そして次の更新伝播送信キュー内の次の要素を読みだし（ステップＳ２４６、ＹＥＳ：ステップＳ２４７）、ステップＳ２４２～Ｓ２４５の処理を繰り返す。

キュー内の全要素に対して処理が完了すると（ステップＳ２４６、ＹＥＳ）、更新伝播送信キューのステップＳ２４５でビットを立てたack 待ちベクタが０にリセットされ、更新要求を送った全てのノードから受信完了の応答があるのを待ってから（ステップＳ２４８）、処理を終了して要求元に復帰する。

【０１３９】

その後変更データ通知部１４は、適当なタイミングで実反映遅延キューを先頭からスキャンし、まだ他ノードに伝播されていない先頭から特定数の変更要求を全アクティブノードに転送する。

[受信データ処理部]

受信データ処理部１５は、他ノードからデータを受信し、自ノードへの反映処理を行う部分である。

【０１４０】

受信データ処理部１５が他ノードから受取るデータには、Read/Write要求、Ｒ

ＥＳＥＴ要求及び等価性回復転送データの４種類があり、受信データ処理部１５はそれぞれに応じた処理を行う。

【０１４１】

図２９は、受信データ処理部１５の動作処理を示すフローチャートである。

受信データ処理部１５は、他ノードから要求を受信すると、まずその内容を判断する（ステップＳ２５１）。その結果更新要求であれば、ステップＳ２５２の更新要求処理を行う。また自ノードがWrite トークンを保持しており、他ノードからRead要求若しくはWrite 要求が送信されてきたのであれば、ステップＳ２５３のRead/Write要求処理を行う。また、他ノードが離脱したノードを検出してＲＥＳＥＴ要求を送信してきたのならば、ステップＳ２５４のＲＥＳＥＴ要求処理を行う。また、自ノードが等価性回復中で、等価性回復転送要求をしたノードから等価性回復転送データを送信してきたのならば、ステップＳ２５５の等価性回復転送データ処理を行う。

【０１４２】

図３０は、図２９のステップＳ２５２の更新要求処理における受信データ処理部１５の処理を示すフローチャートである。

更新要求処理に入ると、受信データ処理部１５は、受信した更新データに対応するオブジェクトグループの内部制御表を参照し、このオブジェクトグループの伝播モードと等価性回復中であるかどうかを調べる。その結果伝播モードが同期モードあるいは半同期モードであるか（ステップＳ２６１、ＹＥＳ）、非同期モードであっても状態フラグに等価性回復中が表示されていた場合（ステップＳ２６１、ＮＯ：ステップＳ２６２、ＹＥＳ）、ＯＳファイルシステムを介して、自ノードの対応ファイルに変更データを直ちに反映させ（ステップＳ２６３）、処理を終了する。

【０１４３】

また転送モードが非同期モードであり（ステップＳ２６１、ＮＯ）、また等価性回復中でなかった場合には（ステップＳ２６２、ＮＯ）、ステップＳ２６２として受信した変更要求を実反映遅延キューの最後尾に繋ぎ、順序性保証を考慮して変更要求を自ファイルへ反映させる。尚順序性保証については後述する。

【０１４４】

図３１は、実反映遅延キューの構成例を示す図である。

実反映遅延キューは、非同期モードによる更新要求をキューイングするバッファで、内部制御表内の実反映遅延キューエントリによって先頭要素の位置が示されるリスト構造を持つキュー部分２１と受信済みベクタ２２によって構成される。キュー部分２１の１つの要素は１つの更新要求に対応しており、受信データ処理部１５は、非同期モードのオブジェクトグループ内のファイルに対する更新要求を受信すると、実反映遅延キューの最後に新規の要素を繋ぎ、処理が完了すると対応する要素を削除する。

【０１４５】

キュー部分２１の１つの要素は、基本的に更新伝播送信キューの要素と同じ構成で、次の要素の位置を示すポインタ、更新を行うファイルが属するオブジェクトグループのオブジェクトグループ番号、この更新要求を他ノードに送信したかどうかを示す送信済みフラグ、各ノード毎の応答状態を示すack 待ちベクタ、更新対象ファイルのファイル名とそのファイル中での更新位置をするオフセット、更新データの大きさを示す長さ、更新要求を行ったノードのノード番号を示す要求ノード番号、更新番号、依存ベクタ、更新内容を示す更新データによって構成される。

【０１４６】

これらのうち更新番号及び依存ベクタは後述する順序性保証処理で用いられるもので、順序性保証の項で詳細に説明する。また送信済みフラグ及びack 待ちベクタは、系構成管理部１１からＲＥＳＥＴ要求を受けた時にのみ用いられる。

【０１４７】

また受信済みベクタ２２は、系内のノード分の要素を備え受信した更新要求内の依存ベクタ最新の依存ベクタが記録される。尚この点についても、順序性保証の項で詳細に説明する。また受信済みマトリックスについても順序性保証の項で説明する。

【０１４８】

図３２は、図２９のステップＳ２５３のRead/Write要求処理における受信デー

タ処理部１５の処理を示すフローチャートである。

Read/Write要求処理に入ると受信データ処理部１５の処理は、受信したRead要求若しくはWrite 要求にオプションでＦＯＲＣＥが指定されているかどうかによって処理が異なる。

【０１４９】

受信したRead/Write要求が、等価性回復中のノードからのものであり、ＦＯＲＣＥオプションの指定されたものである時は（ステップＳ２７１、ＹＥＳ）、ステップＳ２７２としてトークン管理部１３に要求処理に必要なReadトークン若しくはWrite トークンの獲得を依頼する。その結果獲得に成功すれば（ステップＳ２７３、ＹＥＳ）、ステップＳ２７４に処理を移し、獲得に失敗すれば（ステップＳ２７３、ＮＯ）、ステップＳ２７８として要求元ノードにエラー応答を行った後処理を終了する。

【０１５０】

また受信したRead/Write要求が、ＦＯＲＣＥオプションの指定の無いものである時は（ステップＳ２７１、ＮＯ）、自ノードがWrite トークンを保持していない時は（ステップＳ２７９、ＮＯ）、ステップＳ２７８として要求元ノードにエラー応答を行った後処理を終了する。また自ノードがWrite トークンを保持している時は（ステップＳ２７９、ＹＥＳ）、ステップＳ２７４に処理を移す。

【０１５１】

ステップＳ２７４では、内部制御表を参照し、Read/Write要求の対象となっているオブジェクトグループの伝播モードを調べる。その結果同期モードあるいは半同期モードであった場合には（ステップＳ２７４、同期／半同期）、ＯＳのファイルシステムに依頼して、要求された処理を行い（ステップＳ２７６）、結果を要求もとノードに応答して処理を終了する。尚ステップＳ２７６において、Write 要求に対する処理の場合、自ファイルへの書込み処理の他、変更内容の他ノードへの伝播を変更データ通知部１４に依頼する。

【０１５２】

ステップＳ２７４で、Read/Write要求の対象となっているオブジェクトグループの伝播モードが非同期であるならば（ステップＳ２７４、非同期）、後述する

順序性保証の項で述べる順序性保証の為の処理を考慮しつつ、ＩＯ要求インタセプト部１２によるRead/Write要求処理に準じた処理を行い、結果を要求元ノードに返し（ステップＳ２７７）、処理を終了する。

図３３は、図２９のステップＳ２５４のＲＥＳＥＴ要求処理における受信データ処理部１５の処理を示すフローチャートである。

【０１５３】

ＲＥＳＥＴ要求処理に入ると、受信データ処理部１５は、ステップＳ２８１として内部制御表内の実反映遅延キューのエントリから位置を調べ、実反映遅延キューの先頭要素を読み出す。そしてその要素が、

系から離脱したノードからの更新要求を待っているものであるならば（ステップＳ２８２、ＹＥＳ）、ステップＳ２８３としてその更新要求を実反映遅延キューから削除して解放する。また他のノードからの更新要求であったならばそのまま残しておく（ステップＳ２８２、ＮＯ）。

【０１５４】

そして次の実反映遅延キュー内の次の要素を読みだし（ステップＳ２８４、ＮＯ：ステップＳ２８５）、ステップＳ２８２～２８４の処理を繰り返し、キュー内の全要素に対して処理が完了すると（ステップＳ２８４、ＹＥＳ）、処理を終了する。

【０１５５】

図３４は、図２９のステップＳ２５５の等価性回復データ処理における受信データ処理部１５の処理を示すフローチャートである。

等価性回復データ処理に入ると、ステップＳ２９１として受信データ処理部１５は、ステップＳ２９１としてファイルシステムを呼び出し、受信した等価性回復転送データの自ノードのファイルへの反映を依頼し、完了応答を待った後（ステップＳ２９２）、処理を終了する。

［順序性保証］

本システムでは、ファイルの更新を行うと更新内容は更新要求として系内の他ノードに伝播されてゆく。伝播モードとしては、同期、非同期、半同期の３つの

モードがあり、このうち同期モード及び半同期モードによる伝播以外の時は、系の縮退時に完了した筈のファイルの更新の結果が失われてしまう可能性がある。この為、系縮退時に一部データが失われ、結果として新旧データが入り乱れる事態が生じる。半同期モードではしかもファイルへの更新データが他のノードに更新された順番に届くとは限らない。

【０１５６】

本実施形態では、非同期モード時、受信した更新データを実反映遅延キューにキューイングしてゆき、実反映遅延キュー内の更新データの自ファイルへの反映を更新番号と依存ベクタによって管理し、順序性保証を行い、系縮退時に新旧データが入り乱れることを防止する。

【０１５７】

この更新番号と依存ベクタは、例えば内部制御表内に設定される。内部制御表は、オブジェクトグループ毎に展開されるので、この構成の場合、高新番号と依存ベクタもオブジェクトグループ毎に持つことになる。従ってオブジェクトグループを互いに関係があるファイルのみで定義すれば、互いに無関係な更新間の順序性保証は行われず、オーバヘッドを削減することが出来る。

１）更新番号

更新番号は、系内で発生するファイル更新のノード内に閉じた順序性を表す為に単調に増加する番号でありノード毎、オブジェクトグループ毎に用意する。ＩＯ要求インタセプト部１２はユーザプログラムからWrite 要求を受ける度にこの更新番号をインクリメントして更新する。

２）依存ベクタ

依存ベクタは、他ノードの更新番号を含むベクタで、「更新番号で示される更新要求が依存する」他ノードが行った更新を特定する。依存ベクタは、オブジェクトグループ毎に用意され、そのオブジェクトグループに属するノード数個の要素をもつ。

【０１５８】

各要素の内、自ノードに対応する部分には、常に自ノードの更新番号より１つ小さい値が設定される。依存ベクタは、更新データの伝播時に、更新番号共に更

新データに付加されて伝播される。

【０１５９】

Write トークンの獲得に失敗してWrite 処理を他ノードに依頼する場合、ＩＯ要求インタセプト部１２がWrite 要求に更新番号と依存ベクタを付加し、これを依頼先のノードに送信する。Write 要求によるファイルの更新内容は、Write 要求を受けたノード経由で更新伝播時に系内の全ノードに通知される。

【０１６０】

またRead要求を依頼されたノードは、応答も依存ベクタを付加する。

図３５は、Write 要求及びRead要求の応答に付加される依存ベクタの例を示す図である。

【０１６１】

同図上段は３つのノードで系が構成されている場合に、Write 要求ノード２からノード１にWrite 要求を行う場合を示す図であり、下段はRead要求に対する応答を行う場合を示している。

【０１６２】

ノード２のＩＯ要求インタセプト部１２は、ユーザプログラム１７からWrite 要求を受けると、内部制御表内の更新番号及び依存ベクタ内の自己に対応する部分をインクリメントし（同図の場合更新番号を９－＞１０、依存ベクタを（１０、８、６）－＞（１０、９、６）に変更）、これを更新番号と共にWrite 要求に付加してノード１に送る。またRead要求の応答の場合にはこれらの更新は行わず、内部制御表に設定されている依存ベクタをそのまま付加して送信する。

【０１６３】

ノード１では、Write 要求の場合受信した更新データを更新番号及び依存ベクタと共に実反映遅延キューにキューイングすると共に、内部制御表内の依存ベクタと各要素毎に受信済みベクタ２２内のベクタと受信した依存ベクタとを比較（ノード２の部分は更新番号と比較）、受信したベクタの方が大きければこれを新たな値として内部制御表にセットする。

【０１６４】

また図３５下段は、Read要求の応答に対しては、単に内部制御表の内の依存ベ

クタと応答に付加していた依存ベクタとを要素毎に比較し、受信したベクタの方が大きければこれを新たな値として内部制御表にセットする。

【０１６５】

依存ベクタは、他ノードから送信されてきた更新要求や、Write 要求で通知された更新データを実ファイルに反映してもよいかどうかを受信データ処理部１５が判断するのに使用する。受信データ処理部１５は、依存ベクタ内の要素全てのより小さい更新番号の更新要求を各ノードから全て受取済の場合には、実ファイルに反映してよいと判断して更新を行う。

【０１６６】

尚受信した更新要求より先行する更新に対する更新要求にまだ到着していないものが存在する場合、系再構成時の破棄に備え、その未着の更新内容が送られて来るまで受信した更新内容を実反映遅延キューに保持しておき、実ファイルへの反映を遅らせる。これにより、更新内容が前後して届いた場合に系の再構成が生じても、データが破壊されることはない。

【０１６７】

図３６は、受信データ処理部１５が行う依存ベクタによる判断処理を説明する図である。

同図はノード３の実反映遅延キューの状態を示したもので、キューには受信順にノード１からの更新番号１２の更新要求（同図中１／１２）、ノード１からの更新番号１３の更新要求（同図中１／１３）及びノード２からの更新番号１２の更新要求（同図中２／１２）更新要求が実反映遅延キューにキューイングされている。また受信済みベクタ２２から、既に反映済みの更新データとして更新番号がノード１及び２は更新番号１０まで、ノード３は更新番号５までの更新データが自ファイルに反映されていることが判る。

【０１６８】

この状態を初期状態Ｔ０とし、次の状態Ｔ１としてノード２から更新番号の１１の更新要求（依存ベクタ（１０，１０，５））がノード３に到着したとする。

これにより、ノード２からの更新要求は更新番号がまで全て１２揃ったことになるので（受信済みベクタ２２から更新番号１０以前のものは既に反映済み）、

受信済みベクタ２２を（１０，１０，５）から（１０，１２，５）と変更すると共に反映可能となった２／１１の更新データを自ファイルに反映させる。しかし、２／１２の更新データに関しては、２／１２の更新データの依存ベクタと受信済みベクタ２２の値とを比較すると、ノード１の部分の値が２／１２の更新要求の方が大きいので、これは自ファイルには反映させずに実反映遅延キュー内に保持しておく。

【０１６９】

また次のＴ３の状態として、ノード１から更新番号１１の要求（要求１／１１（１０，１１，５）が到着したとする。これによりノード１からの更新要求は更新番号１３まで全てノード３に到着したことになるので、受信済みベクタ２２を（１０，１２，５）から（１３，１２，５）に変更すると共に、反映可能となった要求１／１１，１／１２，１／１３，２／１１を全て実ファイルに反映させ、これらを実反映遅延キューから削除する。

【０１７０】

またRead要求を処理する場合では、実反映遅延キュー対応するデータが退避されていればそちらを優先して読みだし、要求元に送る。この際、応答する依存ベクタもキューイングされているデータに付加されているものを返す。

【０１７１】

この様に処理することにより更新要求が実際の更新順から前後して届いても、受信データ処理部１５は、順序性保ったデータの更新を行うことが出来る。

尚、実反映遅延キューからデータを返す処理を不要にして制御を単純化するために、Write 要求を受取った受信データ処理部１５が、Write 要求に付加された依存ベクタからそのWrite 要求に依存関係の有るデータが自ノードに全て到着するのを待合わせる構成としても良い。この場合Write 要求された更新データの自ファイルへの反映とWrite トークンの解放をそのWrite トークン下で行い、更新を依存するデータが全ノードに到着したことを確認出来るまで遅らせる。この点については後述する。

【０１７２】

この構成の場合、自ノードのデータをRead使用とする場合には、Write トーク

ンの解放を介して、依存するデータが自ノードに反映済みとなるので、Read要求の処理で実反映遅延キューからデータを取り出し応答するという処理が不要となる。ただしこの場合でも、系再編成によりデータの順序性が崩れることを防ぐため、実反映遅延キューを介して、実ファイルのへ反映を遅らせる処理は依然必要となる。

３）依存ベクタの更新タイミング

依存ベクタは以下のタイミングで更新される。

ａ）他ノードからWrite 要求が送られてきた時

受信データ処理部１５は自ノードの依存ベクタの要求元ノードに対応する要素に送られてきた更新番号を設定する。

ｂ）ＩＯ要求インタセプト部１２が他ノードにRead要求を送り、応答としてReadデータをもらった時

受信データ処理部１５は、応答と共に送られてきた依存ベクタと自身が内部制御表内に保持する依存ベクタとを要素毎に比較し、大きい値を内部制御表内に設定する。Read要求を受けたときは受信データ処理部１５は、Read要求を受けた時点の依存ベクタを応答に付加して返す。

【０１７３】

上記の様に依存ベクタを伝播することで、複数のノード間に跨がるデータ間の依存性を表現することが出来る。例えば、ａ(ノード１)->ｂ(ノード２)->ｃ(ノード３)で表現される依存関係がある更新要求の場合、ノード３から送られてきた更新要求ｃは更新要求ａ、ｂの更新が伝播するまで不揮発化が延ばされる。

【０１７４】

図３７は、依存関係のある更新要求の順序性の保証を示す図である。

同図は同一のオブジェクトグループに属するファイルｆａ、ｆｂ及びｆｃの３のファイルに対し３つのノード１、２及び３によってRead/Write要求が発生した場合の依存ベクタによるを示したもので、ｔ０～ｔ５の順でファイルに対する更新が行われた場合、ｔ０、ｔ２、ｔ４の３つの状態で発生した更新要求に付加される依存ベクタには、（０，０，０）＜（１，０，０）＜（１，１，０）の関係が有るので、各ノードに更新要求が順不同で届いてもファイルには順番に反映さ

れる。

４）参照要求時

ユーザプログラム１７からのRead要求に対し、他のノードにRead要求を依頼して応答結果を得る場合、ＩＯ要求インタセプト部１２は、受取ったデータに付加されている依存ベクタで示された受信データに依存関係が有る更新要求を全て受信するまで、ユーザプログラム１７に参照結果を渡さない。

【０１７５】

この様にユーザプログラム１７に応答を返すのを遅らせて、同期を取ることにより、系の再構成を跨がってこのノードが生き続けた場合に、ユーザプログラム１７が参照したデータが失われてユーザプログラム１７の誤動作を防ぐことが出来る。

【０１７６】

尚処理を単純にするため、他ノードにRead要求に対する応答を返す場合、受信データ処理部１５で、自ノードがそれまでに行った変更が過半数のノードに伝わるのを待ってから応答を返すという構成にすることも出来る。この構成の場合には、他ノードにRead要求の応答結果を返す時にはその応答結果が依存する更新要求が系内の過半数のノードに必ず反映済みであることが保証される。よって、更新要求ａ（ノード１）－＞更新要求ｂ（ノード２）－＞更新要求ｃ（ノード３）の様な間接的な依存関係が更新に対しても、ノード２がノード１からReadデータを受信した時点で、更新要求ａが過半数のノードに伝播していることになり、ノード３がノード２からRead結果を受信した時点では依存関係にある更新要求ａが過半数に伝播していることが保証されることになる。

【０１７７】

更に、図３１に示す受信済みマトリックスを導入して、ＷＲＩＴＥトークンの回収をＷＲＩＴＥトークンで保護された更新と依存関係の有る更新要求が全ノードに伝わるまで遅らせる最適化を行う構成とすることも可能である。

【０１７８】

この構成の場合、更新要求は依存関係を持つ更新要求が系内の全ノードに伝播されるまで更新伝播送信キューに繋がれたままとなる。拠って、Read要求に対し

更新伝播送信キューに繋がっていないデータを返す場合には、依存するデータが既に系内の全ノードに伝わっている保証がとれる。

【０１７９】

従って、他ノードからのRead要求に対し、要求を依頼されたノードは更新伝播送信キューにあるデータを応答とするときのみ、その応答としたデータに対応する依存ベクタを応答すればよく、更新伝播送信キューにないデータを応答とする場合には、依存ベクタなしを応答することができる。依存ベクタなしを応答されたＲＥＡＤ要求ノードは依存関係に変更がないので自身の依存ベクタを更新したり、依存ベクタで規定される更新要求が到着するのを待ち合わせる必要がなくなる。

【０１８０】

図３１に示す受信済みマトリックスは、ノード毎に存在するマトリックスで、他ノードの受信済みベクタを要素として持ち、自ノードが認識している他ノードの進行状況を示す。上記したＷＲＩＴＥトークンの回収をＷＲＩＴＥトークンで保護された更新と依存関係の有る更新要求が全ノードに伝わるまで遅らせる構成の場合、ＷＲＩＴＥトークン保持ノードは、この受信済みマトリクスから、依存関係の有る更新が全ノードに伝達されたことを認識する。

【０１８１】

各ノードは一定時間毎に、系内の全ての他ノードに対し自身の受信済みマトリックスをメッセージとして広報し、このメッセージを受信したノードは自身の受信済みマトリックスを更新する。受信マトリクスの更新方法は対応する受信済みベクタに対し、依存ベクタの更新方法で説明したのと同じ方法を適用すればよい。

５）データ更新時

他ノードにWrite 要求を依頼した場合、ＩＯ要求インタセプト部１２は応答で通知される依存ベクタ（更新伝播送信キューに存在する同一ファイルに対する更新の最終要求を示す依存ベクタ）からそれ以前の更新における更新データが全て到着するのを待合わせ、その後自身のデータも更新する。

【０１８２】

Write 要求は自身がそれ以前に行ったRead/Write要求に依存している。このうち、自身のWrite データは上記待合わせ処理により自身で反映済みで有ることが保証される。

【０１８３】

また、参照データに関しては４）で述べた処理により、受取ったデータが依存するデータが全て自ノードに反映済みであることが保証される。従って、Write要求時点で更新要求のデータが依存する他の更新データが自ノードでのファイルに反映済みである保証が得られている。尚更新データを他ノードからの伝播を待たず自ノードに反映しておくのは４）で述べたのと同じ理由で系再編を跨がって動作を続けるユーザプログラム１７の誤動作を防止するためである。

【０１８４】

一方更新データの自ノードへの反映を先に行うと、同じファイルに対する古い伝播が後で到着したり、その更新が前提とする更新が系再編で失われることがある。この事態を防ぐために応答で通知された依存ベクタの最大のものを使い、依存関係のある更新データを待合わせる必要がある。

【０１８５】

図３８は他ノードのWrite 要求を処理する時において、更新伝播送信キューに同じファイルに対する更新要求が存在していた場合の処理を説明する図である。

更新伝播送信キューが同図の状態で、ファイルｆａに対するWrite 要求を受けると、受信データ処理部１５は同じファイルｆａに対する最遅の更新要求（要求２／１２）に対応する（１１，１２，６）を依存ベクタとして応答する。もし、更新伝播遅延キューに同じファイルに対する要求が存在しなければ、依存ベクタ無しを応答する。

【０１８６】

図３９は、本実施形態における上記ファイルレプリケーション制御をコンピュータプログラムにより実現した場合の各ノードの構成を示す図である。

各ノードは図３９の様にＣＰＵ３１、ＲＯＭ、ＲＡＭによる主記憶装置３２、補助記憶装置３３（図４のローカルディスク装置に対応）、ディスプレイ、キーボード等の入出力装置（Ｉ／Ｏ）３４、ＬＡＮやＷＡＮ、一般回線等により他ノ

ードとネットワーク接続を行うモデム等のネットワーク接続装置３５及びディスク、磁気テープなどの可搬記録媒体３７から記憶内容を読み出す媒体読取り装置３６を有し、これらが互いにバス３８により接続される構成を備えている。

【０１８７】

また図３９の情報処理システムでは、媒体読取り装置３６により磁気テープ、フロッピーディスク、ＣＤ－ＲＯＭ、ＭＯ等の記録媒体３７に記憶されているプログラム、データを読み出し、これを主記憶装置３２またはハードディスク３３にダウンロードする。そして本実施形態による各処理は、ＣＰＵ３１がこのプログラムやデータを実行することにより、ソフトウエア的に実現することが可能である。

【０１８８】

また、このノードでは、フロッピーディスク等の記録媒体３７を用いてアプリケーションソフトの交換が行われる場合がある。よって、本発明は、計測制御用監視端末やその画面の操作方法に限らず、コンピュータにより使用されたときに、上述の本発明の実施の形態の機能をコンピュータに行わせるためのコンピュータ読み出し可能な記録媒体３７として構成することもできる。

【０１８９】

この場合、「記録媒体」には、例えば図４０に示されるように、ＣＤ－ＲＯＭ、フロッピーディスク（あるいはＭＯ、ＤＶＤ、リムーバブルハードディスク等であってもよい）等の媒体駆動装置４７に脱着可能な可搬記録媒体４６や、ネットワーク回線４３経由で送信される外部の装置（サーバ等）内の記憶手段（データベース等）４２、あるいは情報処理装置４１の本体４４内のメモリ（ＲＡＭ又はハードディスク等）４４等が含まれる。可搬記録媒体４６や記憶手段（データベース等）４２に記憶されているプログラムは、本体４４内のメモリ（ＲＡＭ又はハードディスク等）４５にロードされて、実行される。

【０１９０】

【発明の効果】

本発明によれば、共用ファイルへのアクセス要求が生じたノードに対し、その共用ファイルに対する最新のデータを保持するノードが通知される。よって、共

用ファイルをアクセスするノードは常に最新のデータに対してアクセスすることが出来る。また各ノードは同一のデータを参照することになるので、各ノードからは一貫性の有るデータが見える。

【０１９１】

また各ノードは、トークンの獲得に失敗してもトークンを獲得できるまで待つことなく処理を続行できる。更に複数のノードによる同一のファイルに対する同時アクセスを可能としている。この為、高い反応性を持つシステムを構築することができる。

ータが見える。

【０１９２】

更に更新内容を他ノードに非同期で伝送しても、全ノードから同じデータが見える。

また更新データには更新の順序性、依存性を示す情報が付加されており、この情報に基づいてファイルの更新が行われるので、途中で系の再構成が生じても、データ更新の順序性が壊れることはない。また動作中の他ノードから矛盾したデータが見えることはない。

【０１９３】

更に、１乃至複数のファイル毎に更新内容の伝播方式や伝播させるノードを設定できるので、業務の性格や性能要件に基づいて設定を行える。

また、新規ノードの参加時において、最新データの復元処理中に生じたアクセス要求を最新データを保持している他のノードに送ることにより、復元処理の完了を待たずに新規参加ノードの業務を開始することが出来る。更に、この時、系内で復元処理と平衡して現在系に加わっているノードの業務を続行できる。

【０１９４】

又共用ファイルに対する処理を、該共用ファイルを共用する他ノードと同期して停止する整然停止を行った場合、共用ファイルへの処理を再開する際、他ノードと同期して再開することにより共用ファイルに対するデータの復元処理を行う必要が無い。

【図面の簡単な説明】

【図１】

本発明の原理図である。

【図２】

系の構成を示す図である。

【図３】

本発明における基本原理を示す図である。

【図４】

本実施形態の系を構成するノードの構成を示すブロック図である。

【図５】

系状態テーブルの構成例を示す図である。

【図６】

内部制御表の構成例を示す図である。

【図７】

Joinコマンド投入時の系構成管理部による動作処理を示すフローチャートである。

【図８】

参入処理時の系構成管理部の動作処理を示すフローチャートである。

【図９】

ＪＯＩＮ要求受付処理時の系構成管理部の動作処理を示すフローチャートである。

【図１０】

Join通知を受取ったノードの系構成管理部が行う処理を示すフローチャートである。

【図１１】

等価性回復処理の系構成管理部の動作処理を示すフローチャートである。

【図１２】

等価性回復転送要求を受信したノードの系構成管理部が行う動作処理を示すフローチャートである。

【図１３】

等価性回復完了メッセージを受信したノードの系構成管理部が行う動作処理を示すフローチャートである。

【図１４】

leave コマンドを投入された時の時の系構成管理部の動作処理を示すフローチャートである。

【図１５】

系内の他ノードの離脱を認識したノードの系構成管理部の処理動作を示すフローチャートである。

【図１６】

ＩＯ要求インタセプト部による処理動作を示すフローチャートである。

【図１７】

トークン制御表の構成例を示す図である。

【図１８】

トークン管理ノードのトークン管理部の処理動作を示すフローチャートである。

【図１９】

Write トークン獲得要求処理時のトークン管理部の処理動作を示すフローチャートである。

【図２０】

Readトークン獲得要求処理時のトークン管理部の処理動作を示すフローチャートである。

【図２１】

トークン解放／回収要求処理時のトークン管理部の処理動作を示すフローチャートである。

【図２２】

不要になったトークンを自発的に返却しない構成の場合に発行されるWrite トークン回収要求を受けたWrite トークン保持ノードが行う動作処理を示すフローチャートである。

【図２３】

変更データ通知部による動作処理を示すフローチャートである。

【図２４】

ＩＯ要求インタセプト部／受信データ処理部呼び出し処理の変更データ通知部の動作処理を示すフローチャートである。

【図２５】

ＳＹＮＣ要求処理時の変更データ通知部の動作処理を示すフローチャートである。

【図２６】

更新伝播送信キューの構成例を示す図である。

【図２７】

ＲＥＳＥＴ要求処理時の変更データ通知部の動作処理を示すフローチャートである。

【図２８】

ＦＳＹＮＣ要求処理時の変更データ通知部の動作処理を示すフローチャートである。

【図２９】

受信データ処理部の動作処理を示すフローチャートである。

【図３０】

更新要求処理における受信データ処理部の動作処理を示すフローチャートである。

【図３１】

実反映遅延キューの構成例を示す図である。

【図３２】

Read/Write要求処理における受信データ処理部の処理を示すフローチャートである。

【図３３】

ＲＥＳＥＴ要求処理における受信データ処理部の動作処理を示すフローチャートである。

【図３４】

等価性回復データ処理における受信データ処理部１５の動作処理を示すフローチャートである。

【図３５】

Write 要求及びRead要求の応答に付加される依存ベクタの例を示す図である。

【図３６】

依存関係のある更新要求の順序性の保証を示す図である。

【図３７】

依存関係のある更新要求の順序性の保証を示す図である。

【図３８】

Write 要求を自ノードで処理する時において、実反映遅延キューに同じファイルに対する更新要求が存在していた場合の処理を説明する図である。

【図３９】

ノードとなる計算機システムの環境図である。

【図４０】

記憶媒体の例を示す図である。

【符号の説明】

Ａ～Ｊ　ノード

１１　系構成管理部

１２　ＩＯ要求インタセプト部

１３　トークン管理部

１４　変更データ通知部

１５　受信データ処理部

２１　キュー部分

２２　受信済みベクタ

３１　ＣＰＵ

３２　主記憶装置

３３　補助記憶装置

３４　入出力装置

３５　ネットワーク接続装置

３６　媒体読取り装置

３７　可搬記憶媒体

３８　バス

４１　情報処理装置

４２　記憶手段

４３　ネットワーク回線

４４　情報処理装置本体（コンピュータ）

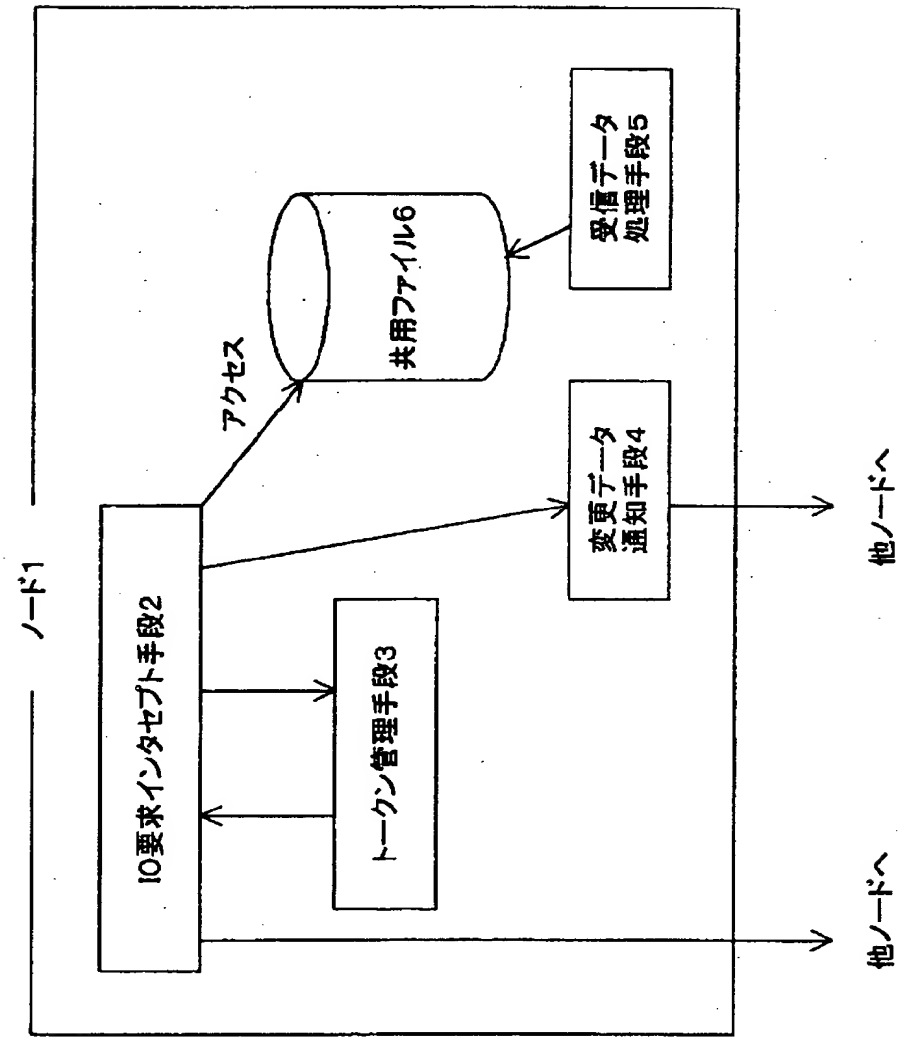
４５　メモリ

４６　可搬記録媒体

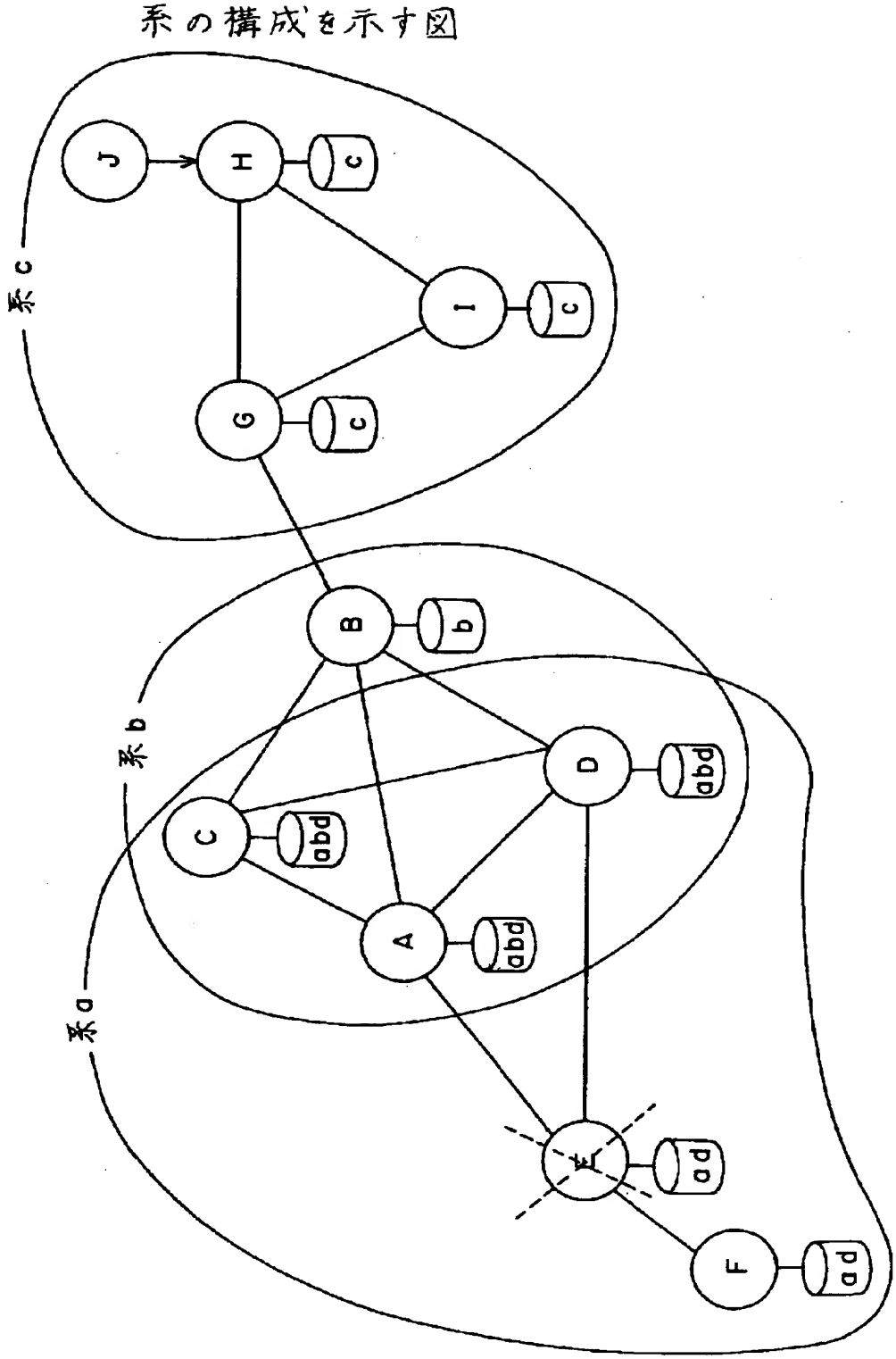
【書類名】　　図面

【図１】

本発明の原理図

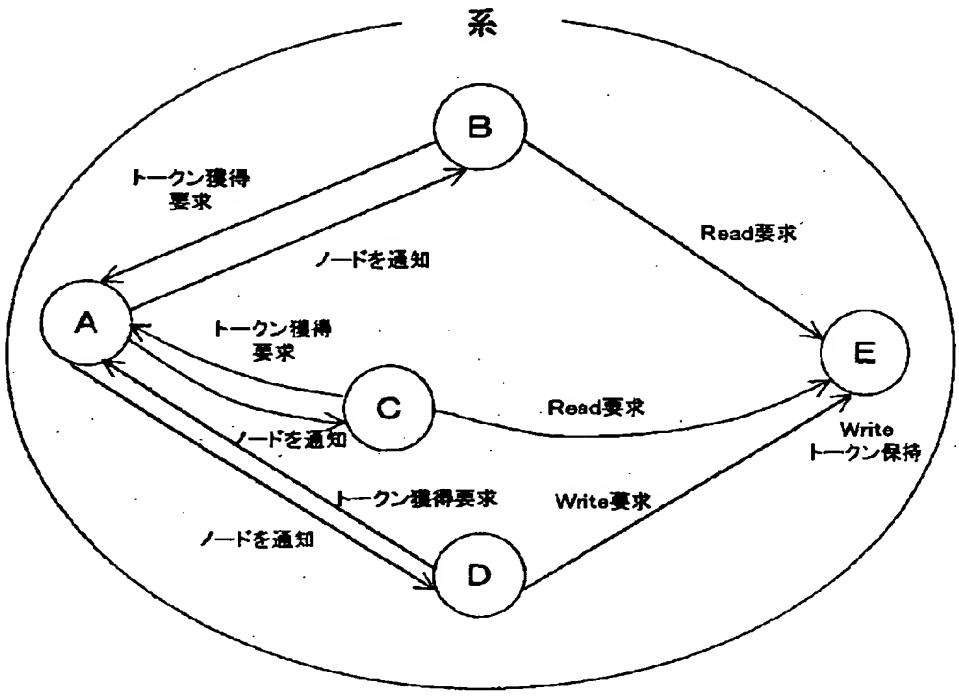


【図２】

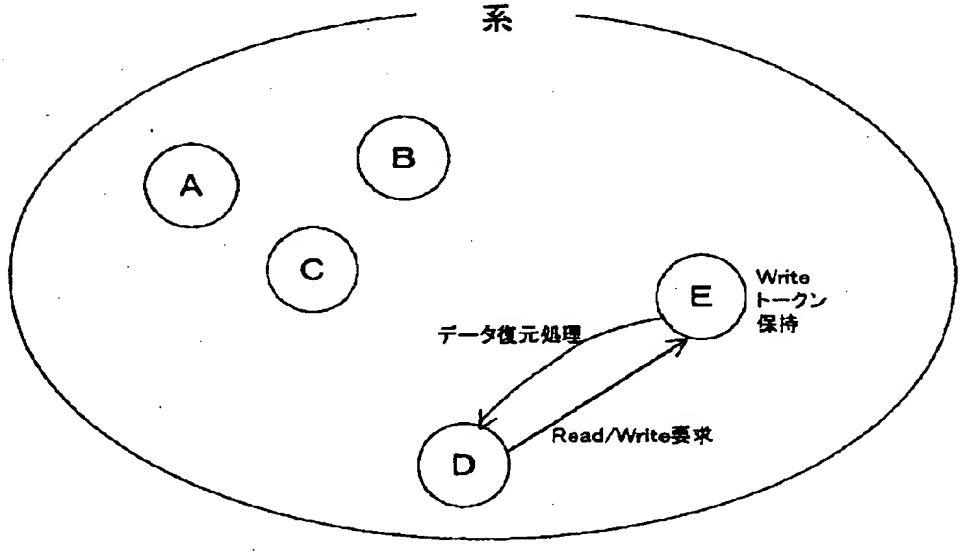


【図３】

本発明における基本原理を示す図



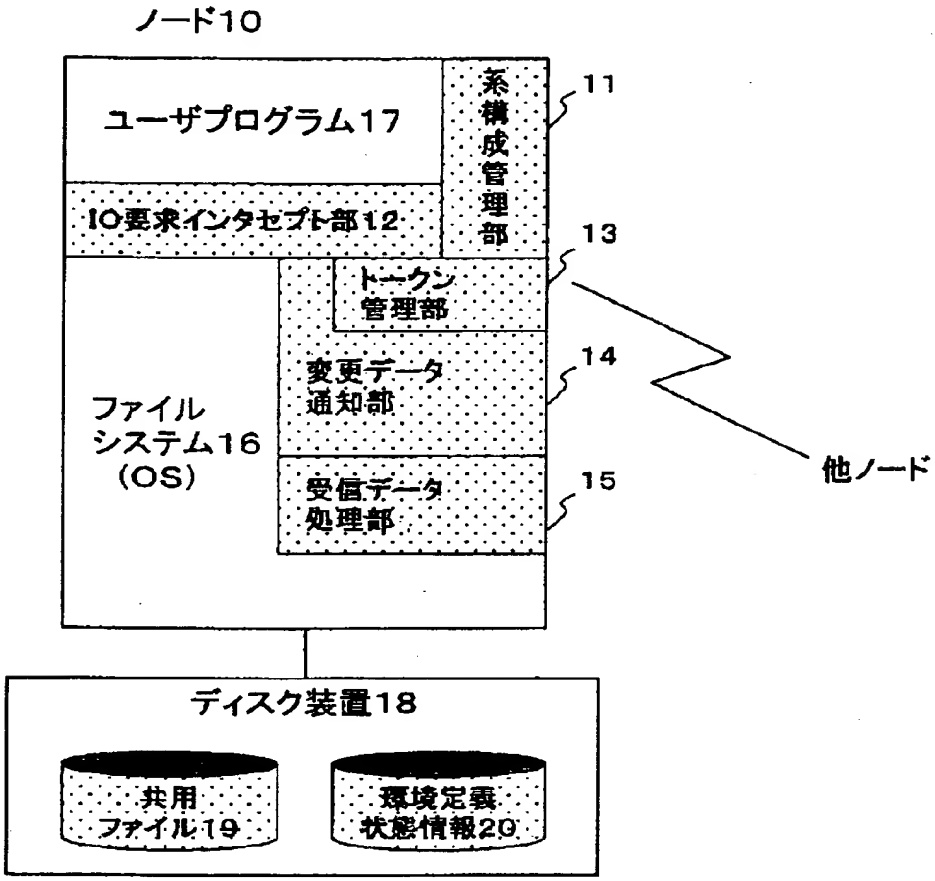
(a)



(b)

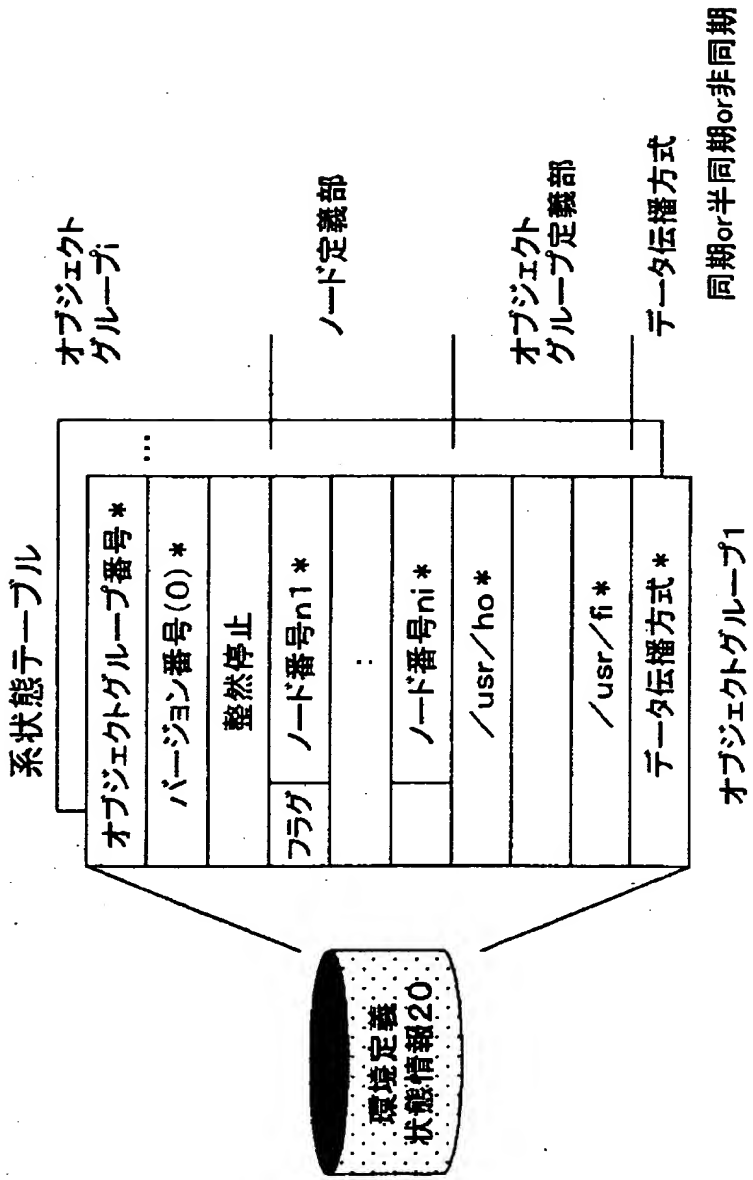
【図４】

本実施形態の系を構成するノードの構成を示すブロック図



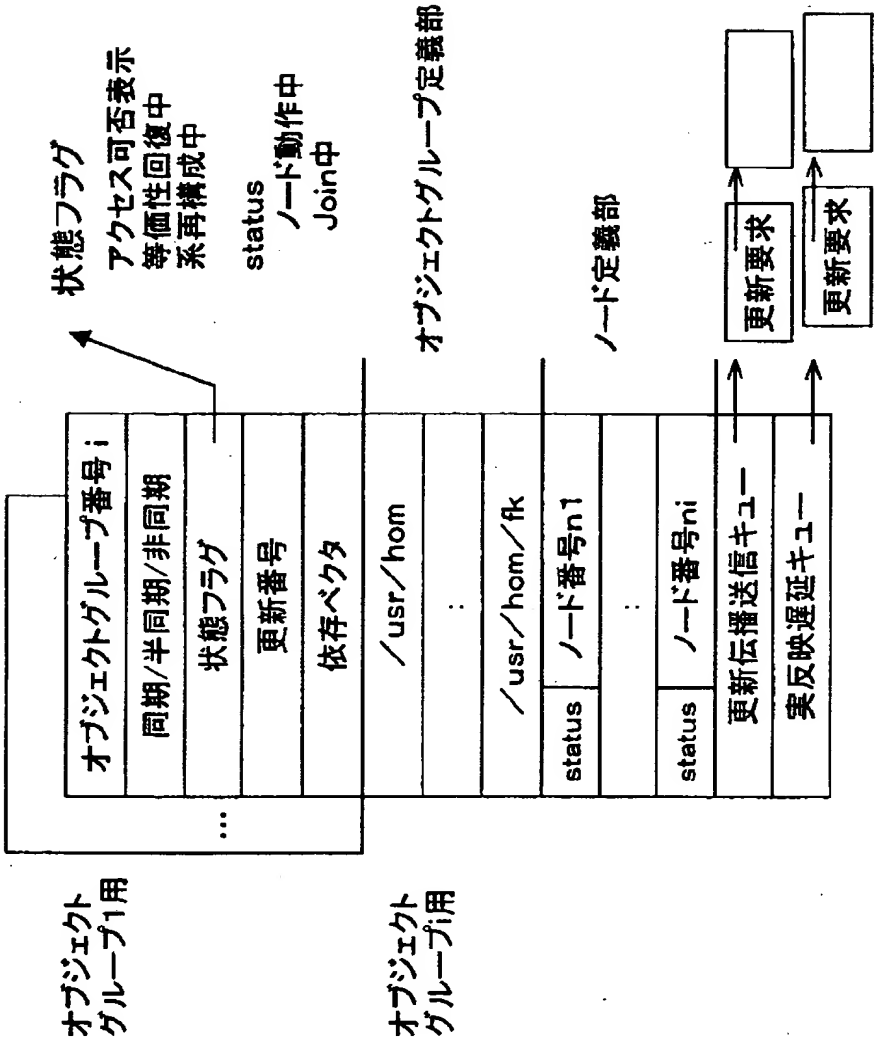
【図５】

系状態テーブルの構成例を示す図



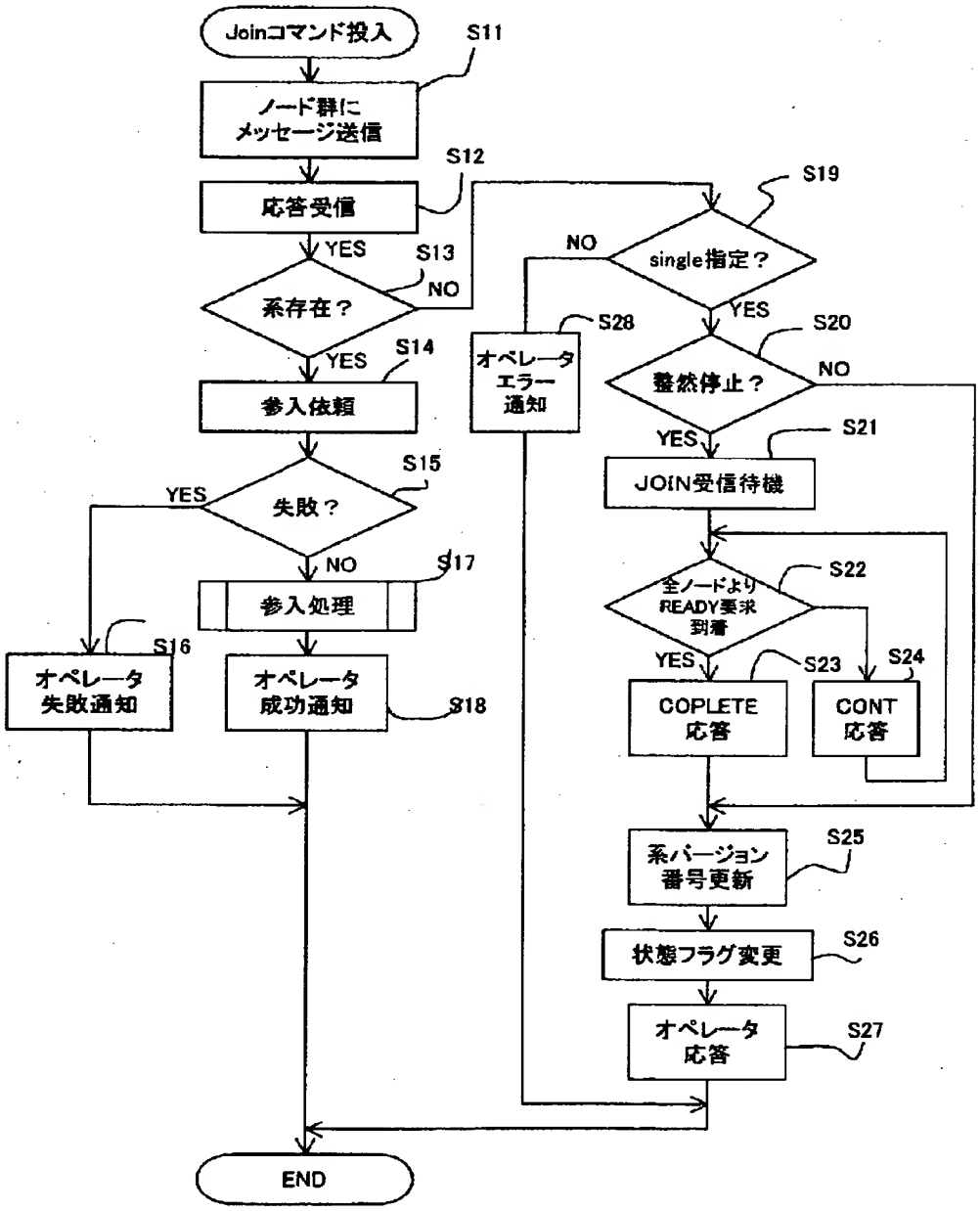
【図６】

内部制御表の構成例を示す図



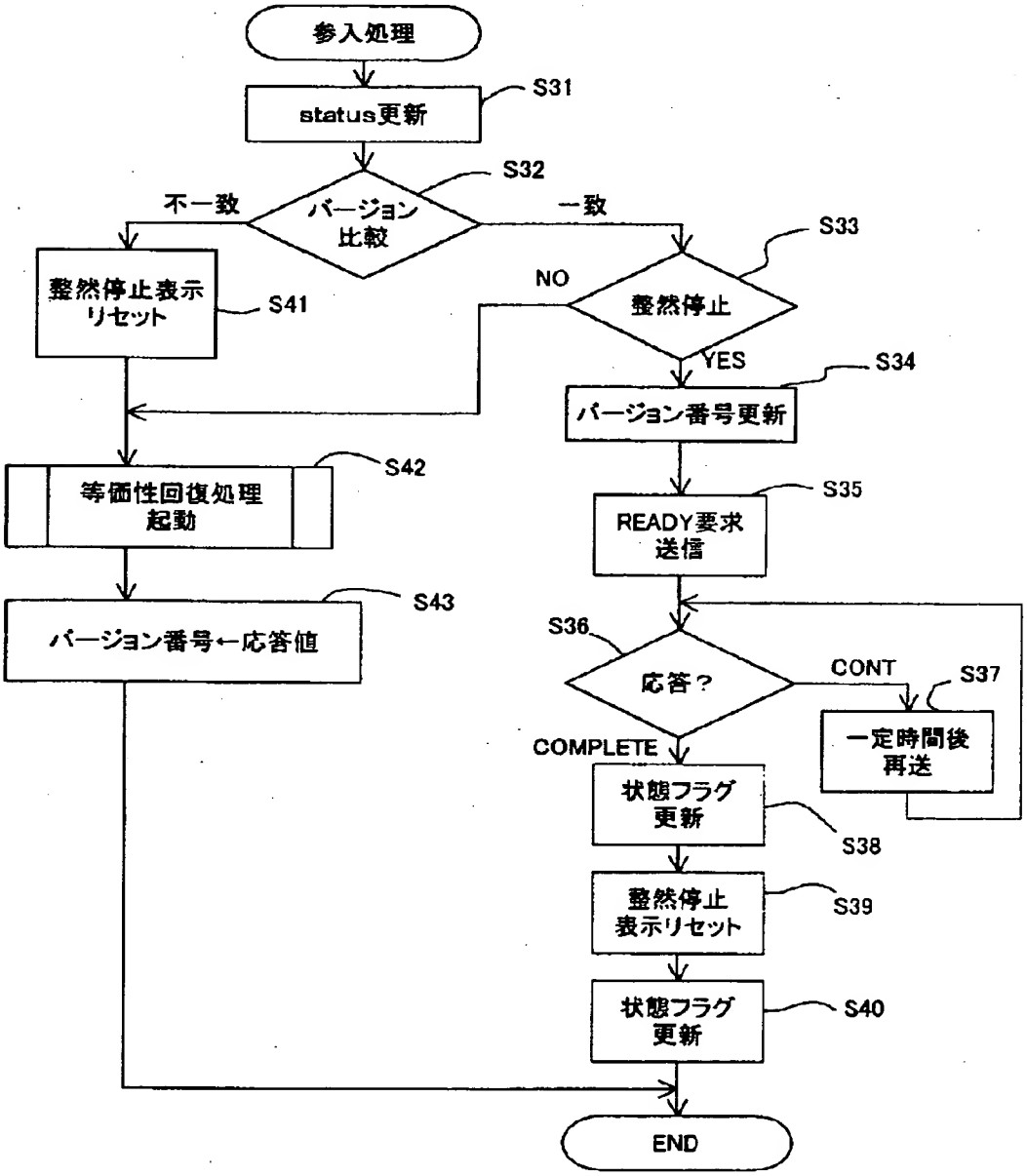
【図７】

Joinコマンド投入時の系構成管理部による
動作処理を示すフローチャート



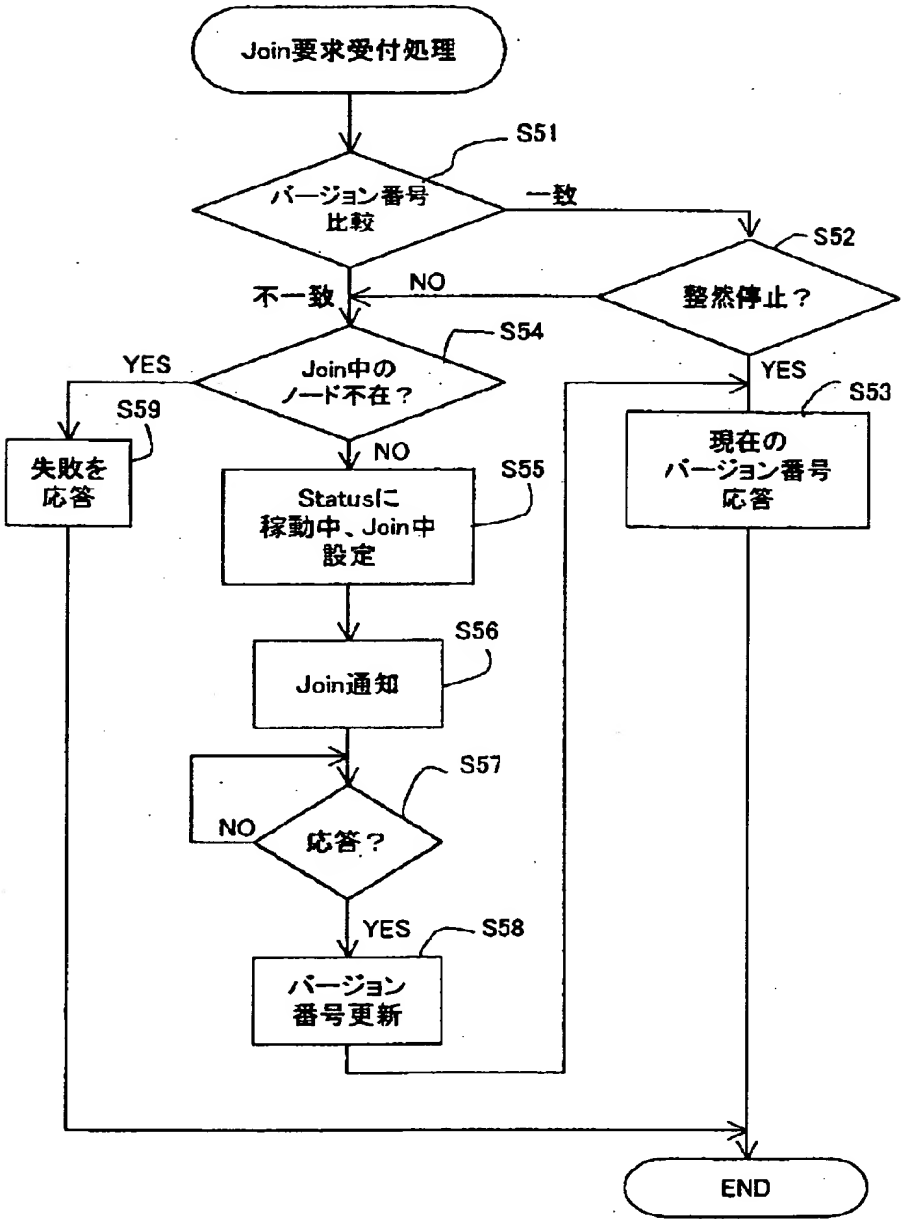
【図８】

参入処理時の系構成管理部の動作処理を示すフローチャート



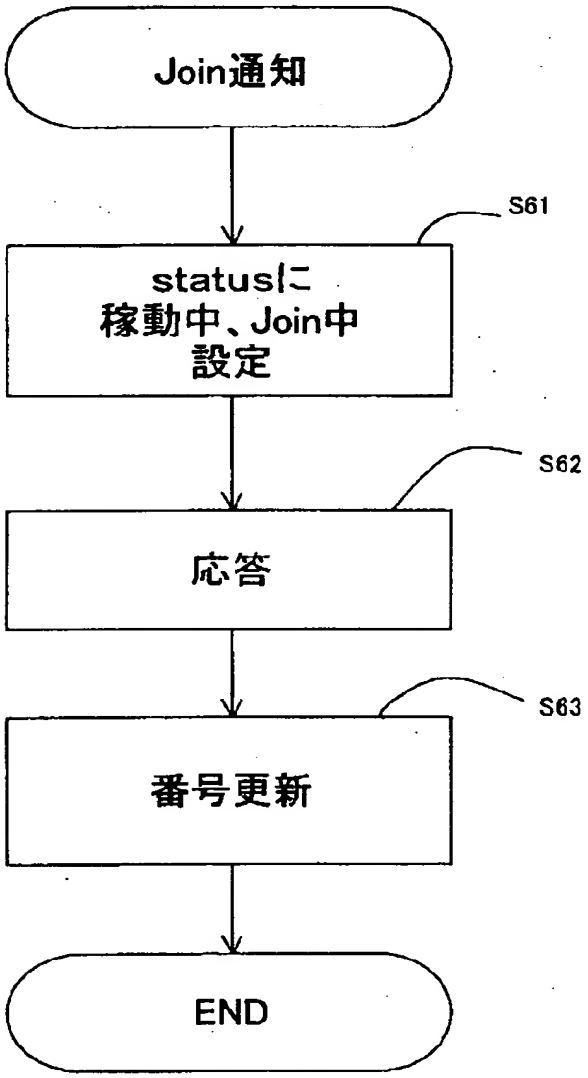
【図９】

JOIN要求受付処理時の系構成管理部の
動作処理を示すフローチャート



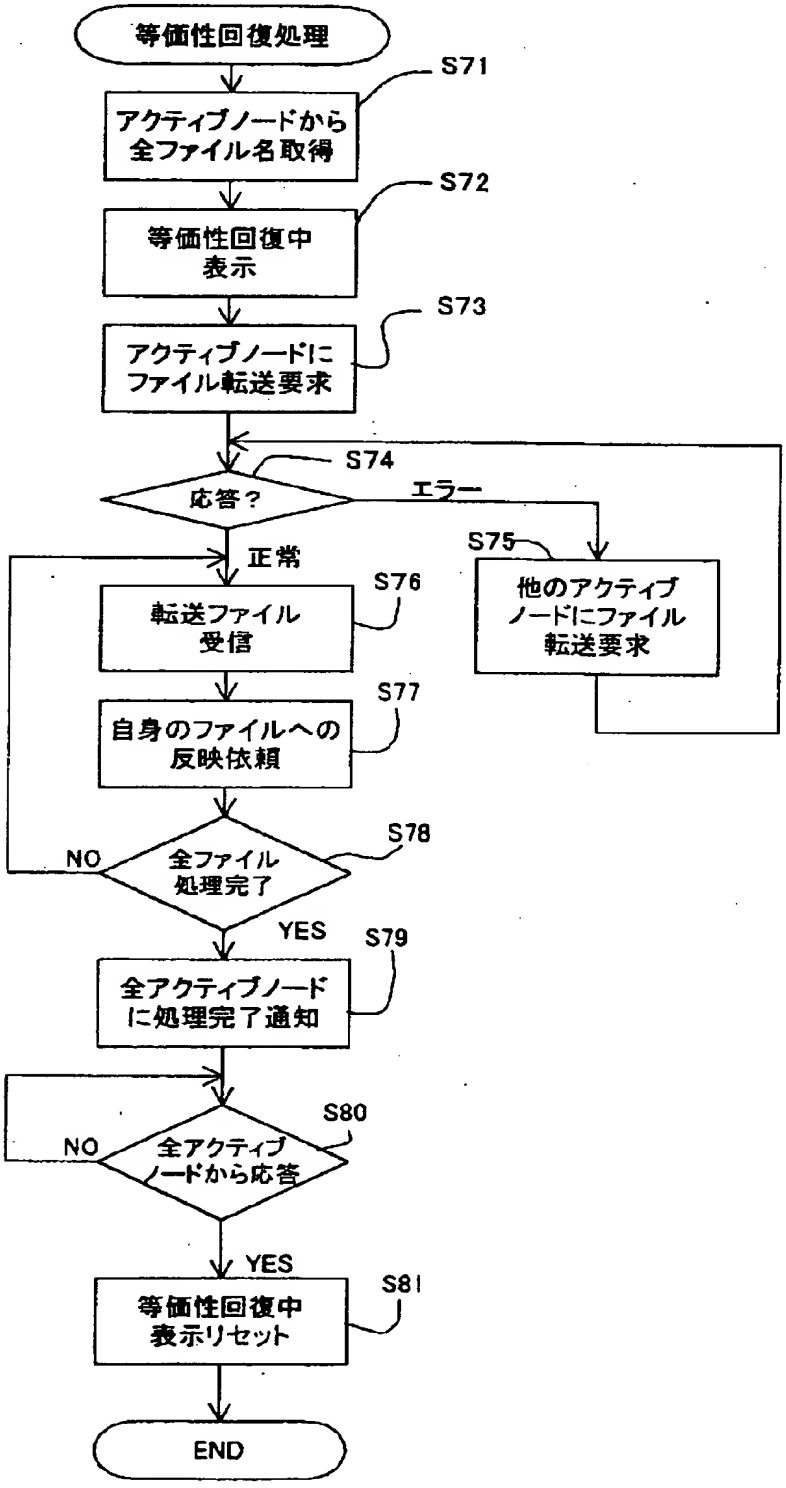
【図１０】

Join通知を受取ったノードの系構成管理部が行う処理を示すフローチャート



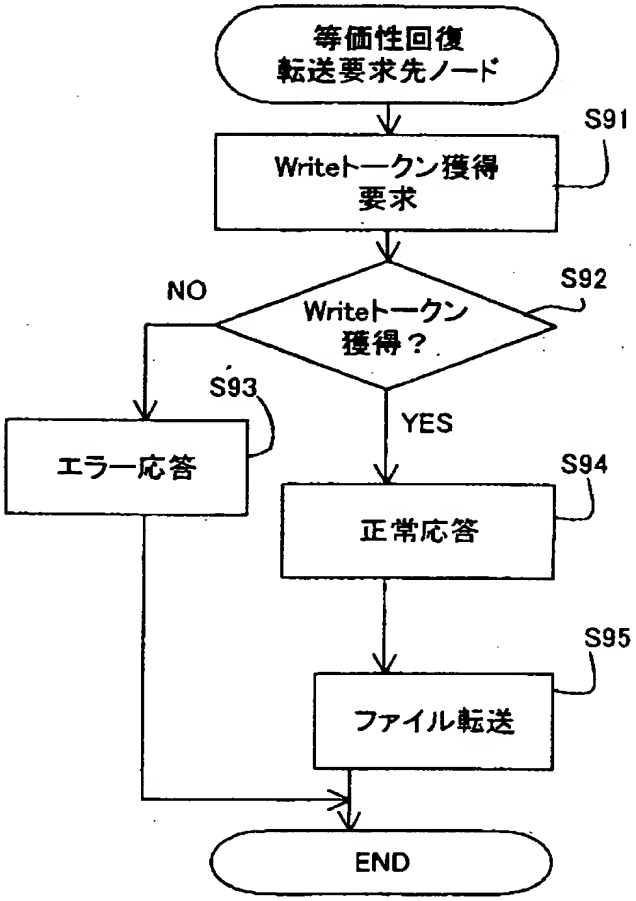
【図１１】

等価性回復処理の系構成管理部の動作処理を示すフローチャート



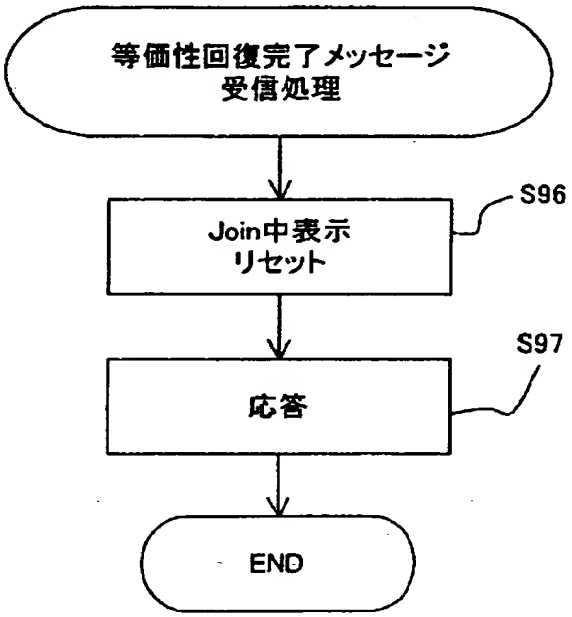
【図１２】

等価性回復転送要求を受信した
ノードの系構成管理部が行う
動作処理を示すフローチャート



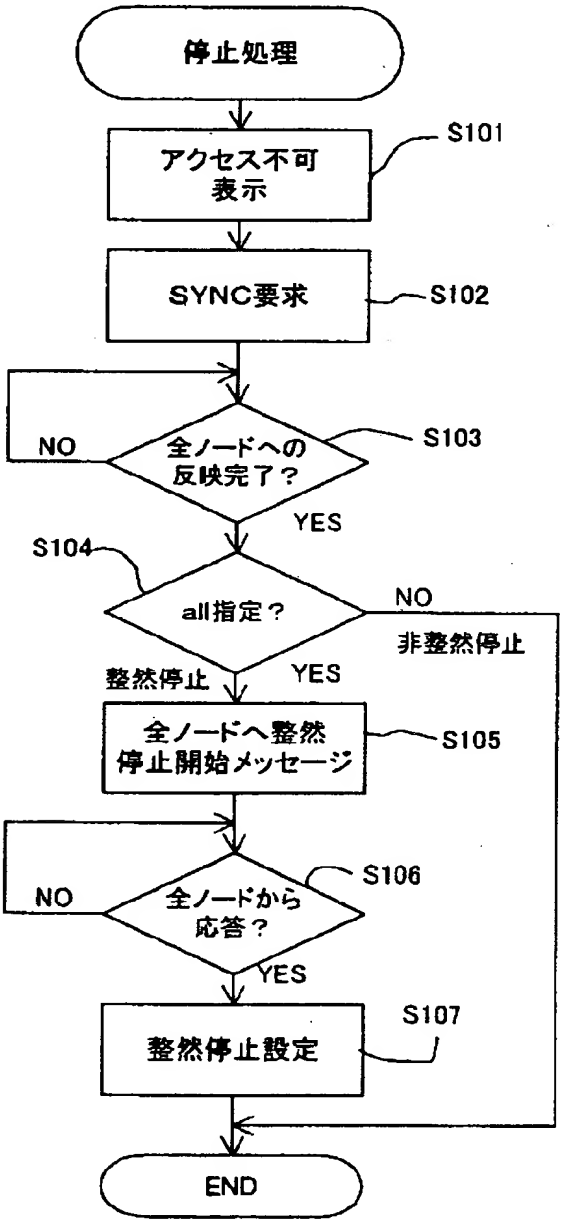
【図１３】

等価性回復完了メッセージを受信したノードの系構成管理部が行う動作処理を示すフローチャート



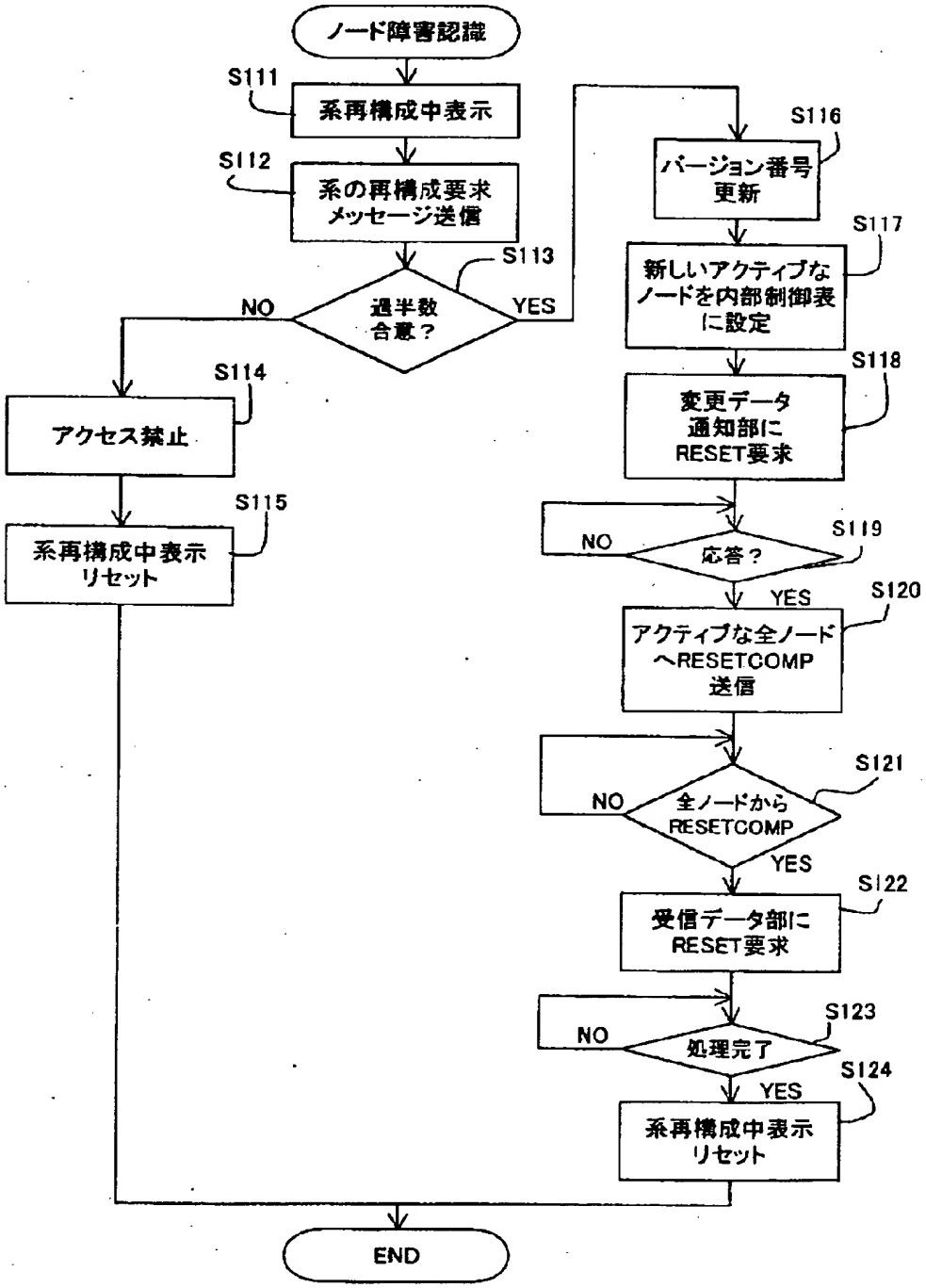
【図１４】

leaveコマンドを投入された時の
系構成管理部の動作処理を示すフローチャート



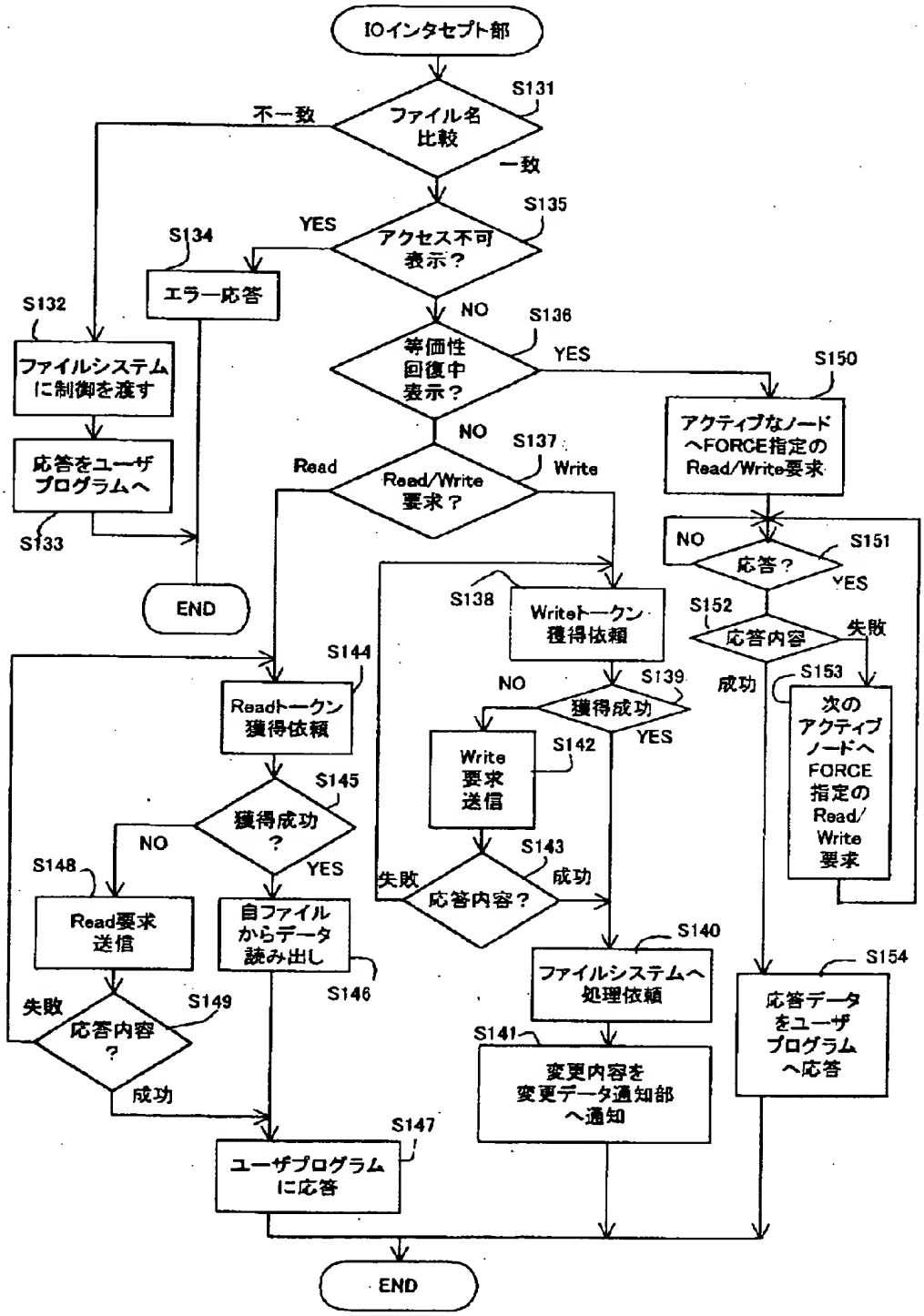
【図１５】

系内の他ノードの離脱を認識したノードの
系構成管理部の処理動作を示すフローチャート



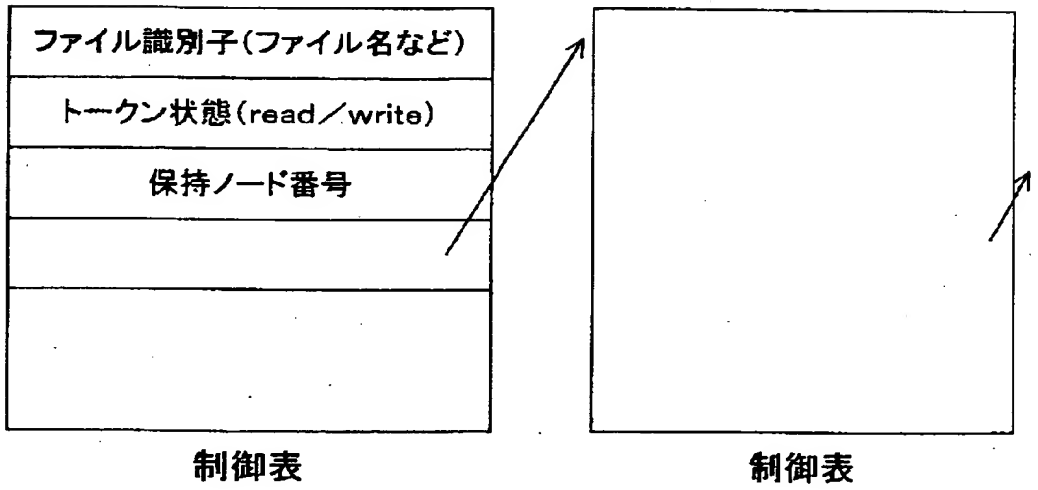
【図１６】

IO要求インタセプト部による処理動作を示すフローチャート



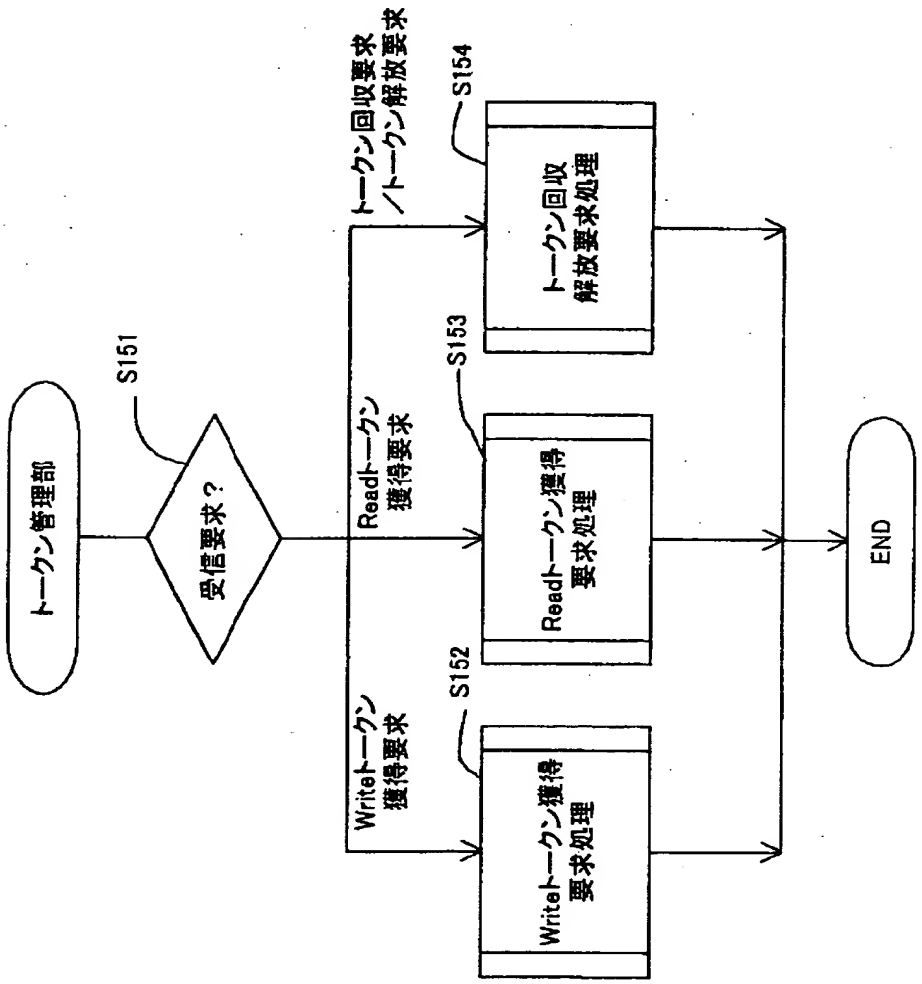
【図１７】

トークン制御表の構成例を示す図



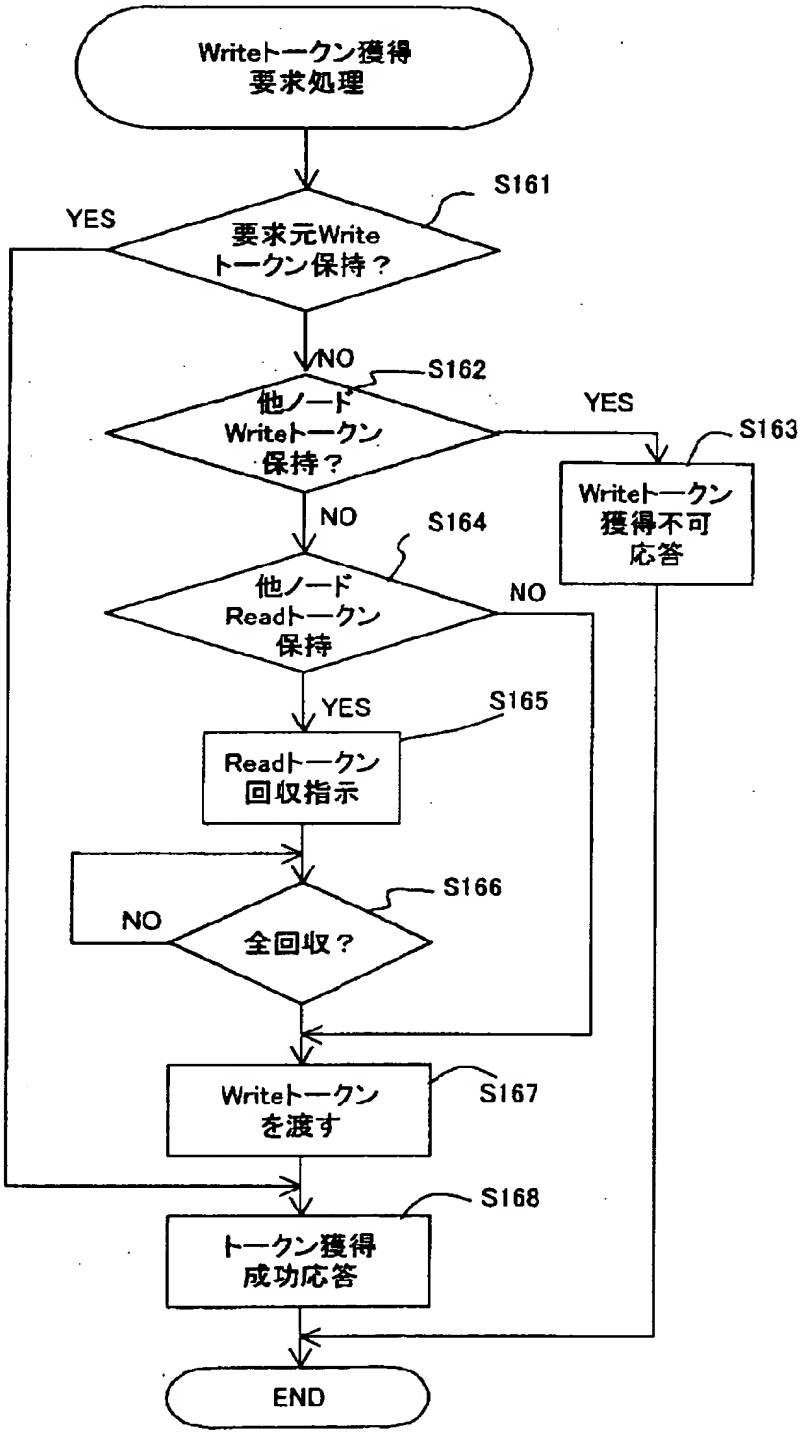
【図１８】

トークン管理部ノードのトークン管理部の処理動作を示すフローチャートである



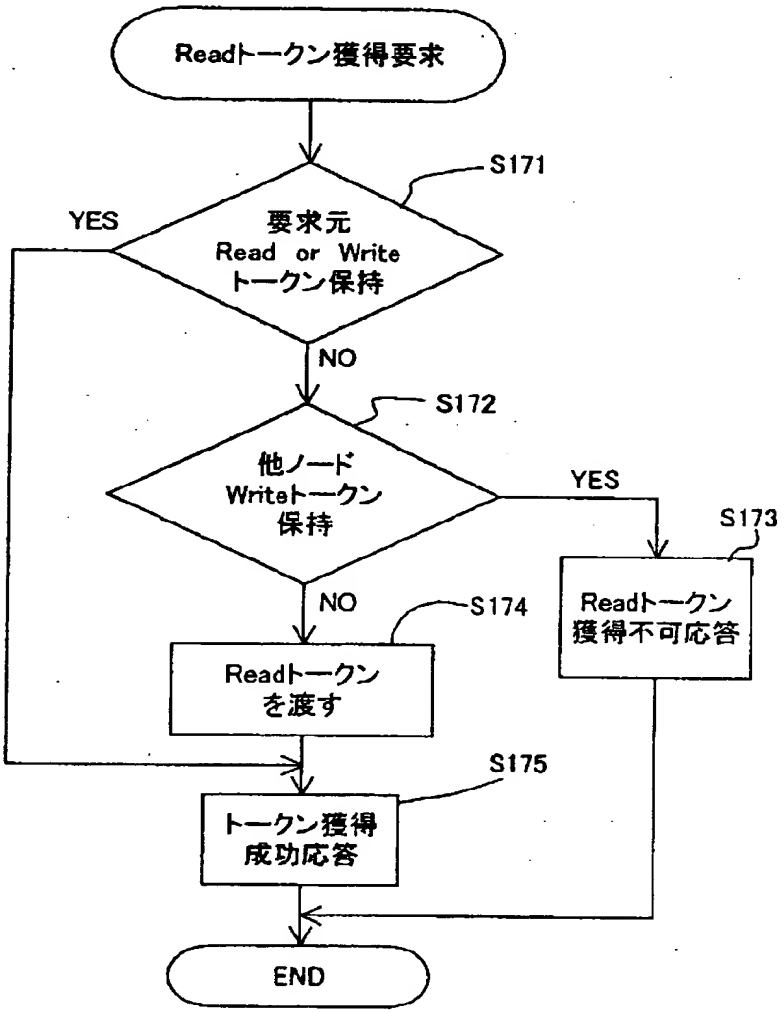
【図１９】

Writeトークン獲得要求処理時のトークン管理部の処理動作を示すフローチャート



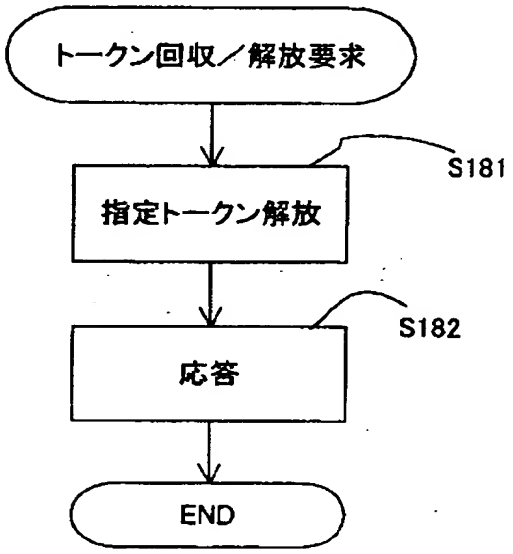
【図２０】

Readトークン獲得要求処理時のトークン管理部の処理動作を示すフローチャート



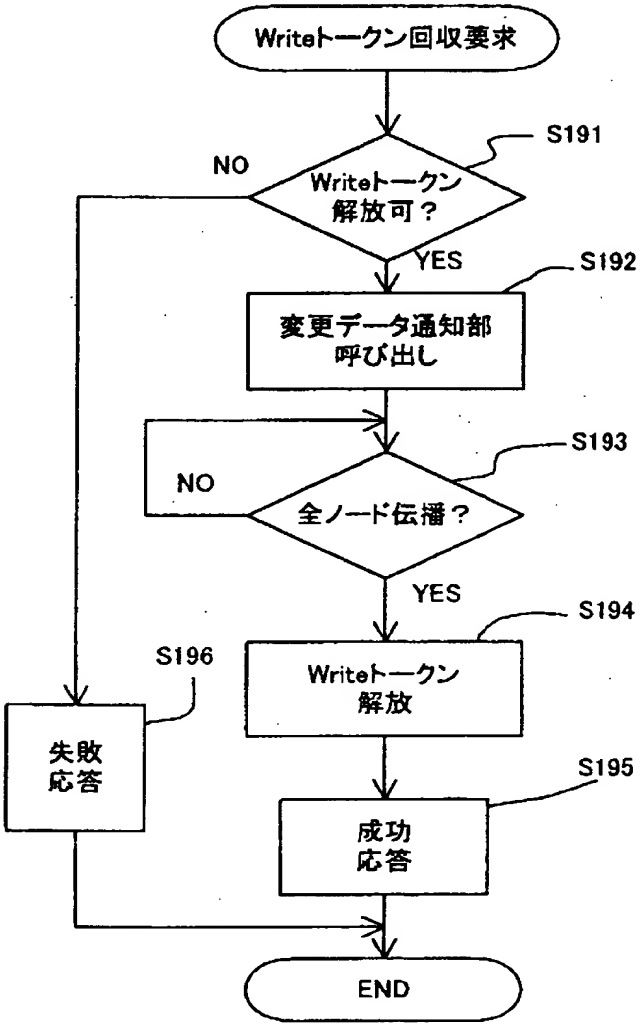
【図２１】

トークン解放／回収要求処理時のトークン管理部の処理動作を示すフローチャート



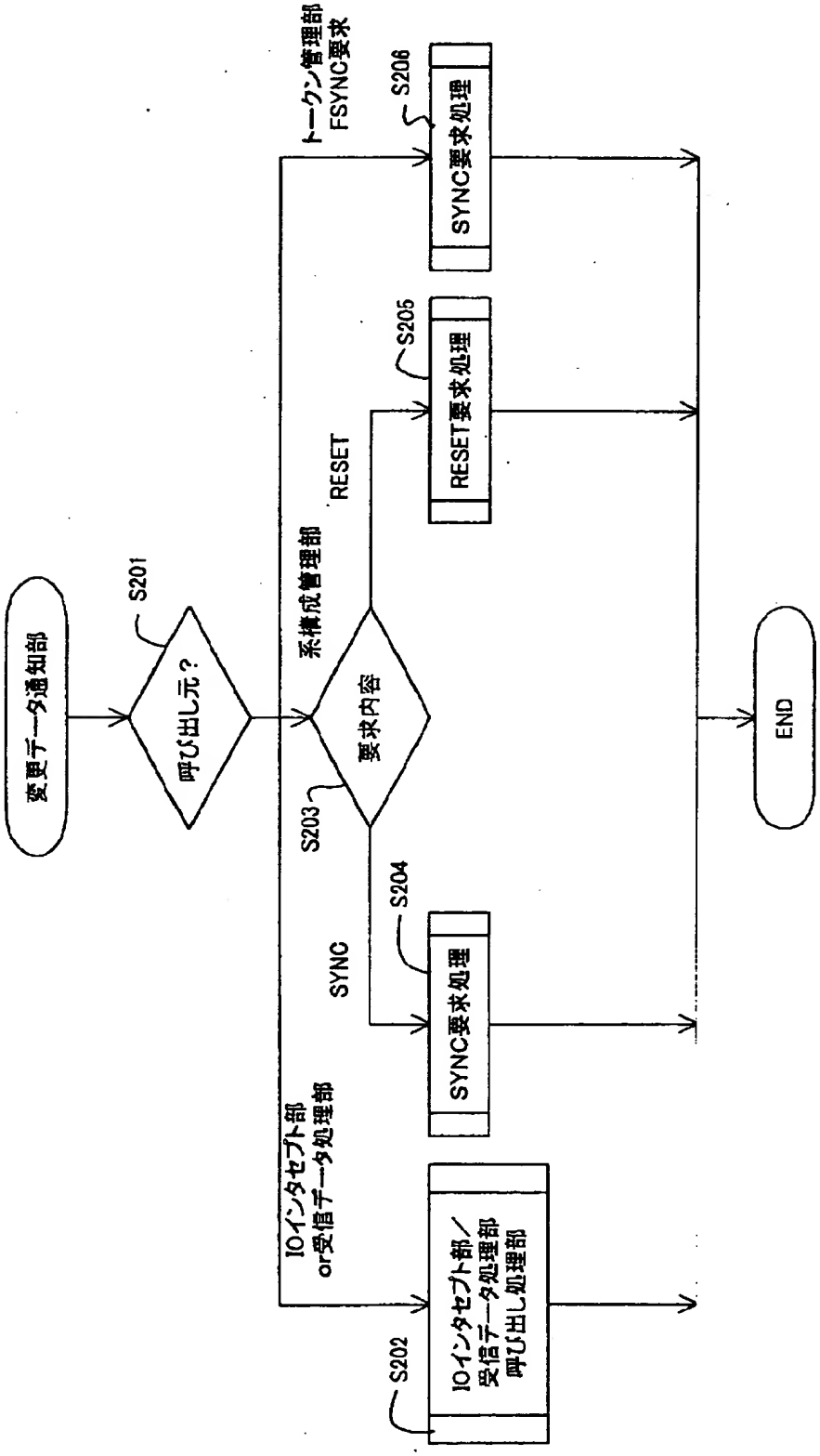
【図２２】

Writeトークン回収要求を受けたWriteトークン保持ノードが行う動作処理を示すフローチャート



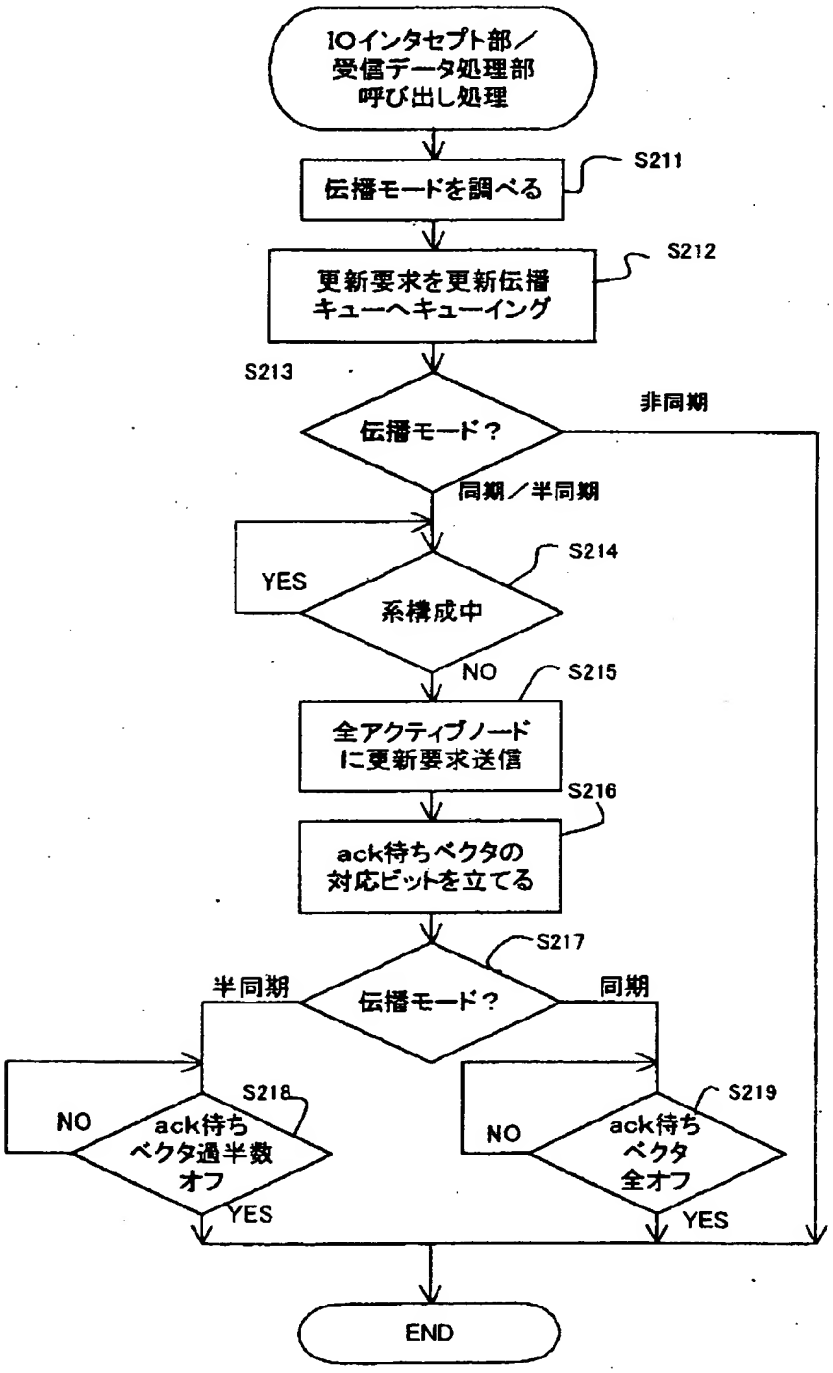
【図２３】

変更データ通知部による動作処理を示すフローチャート



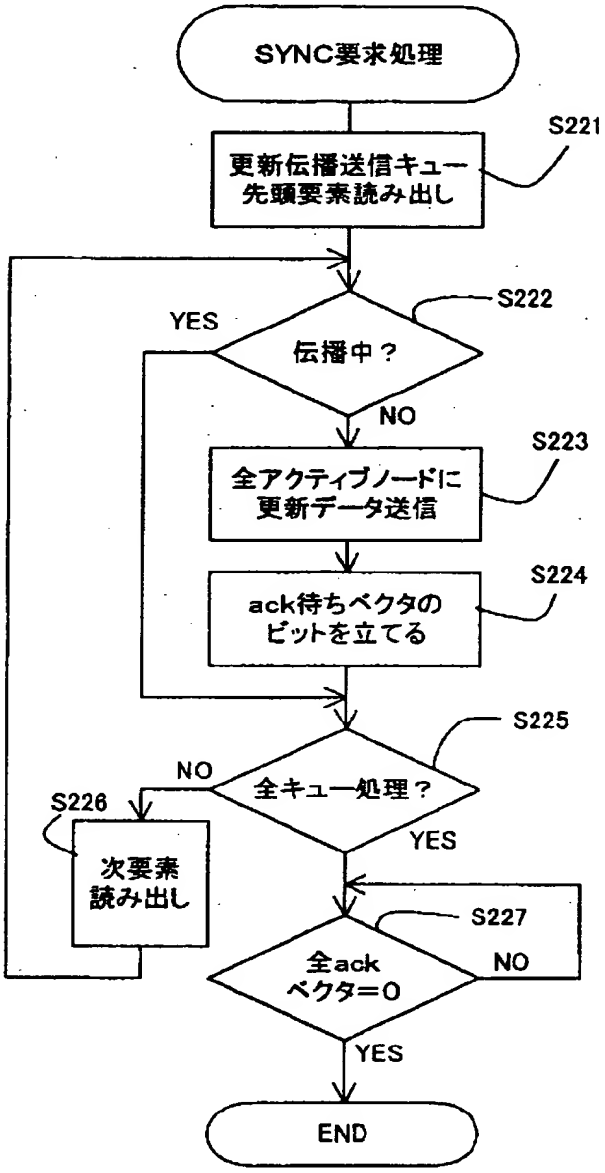
【図２４】

IO要求インタセプト部／受信データ処理部
呼び出し処理の変更データ通知部
の動作処理を示すフローチャート



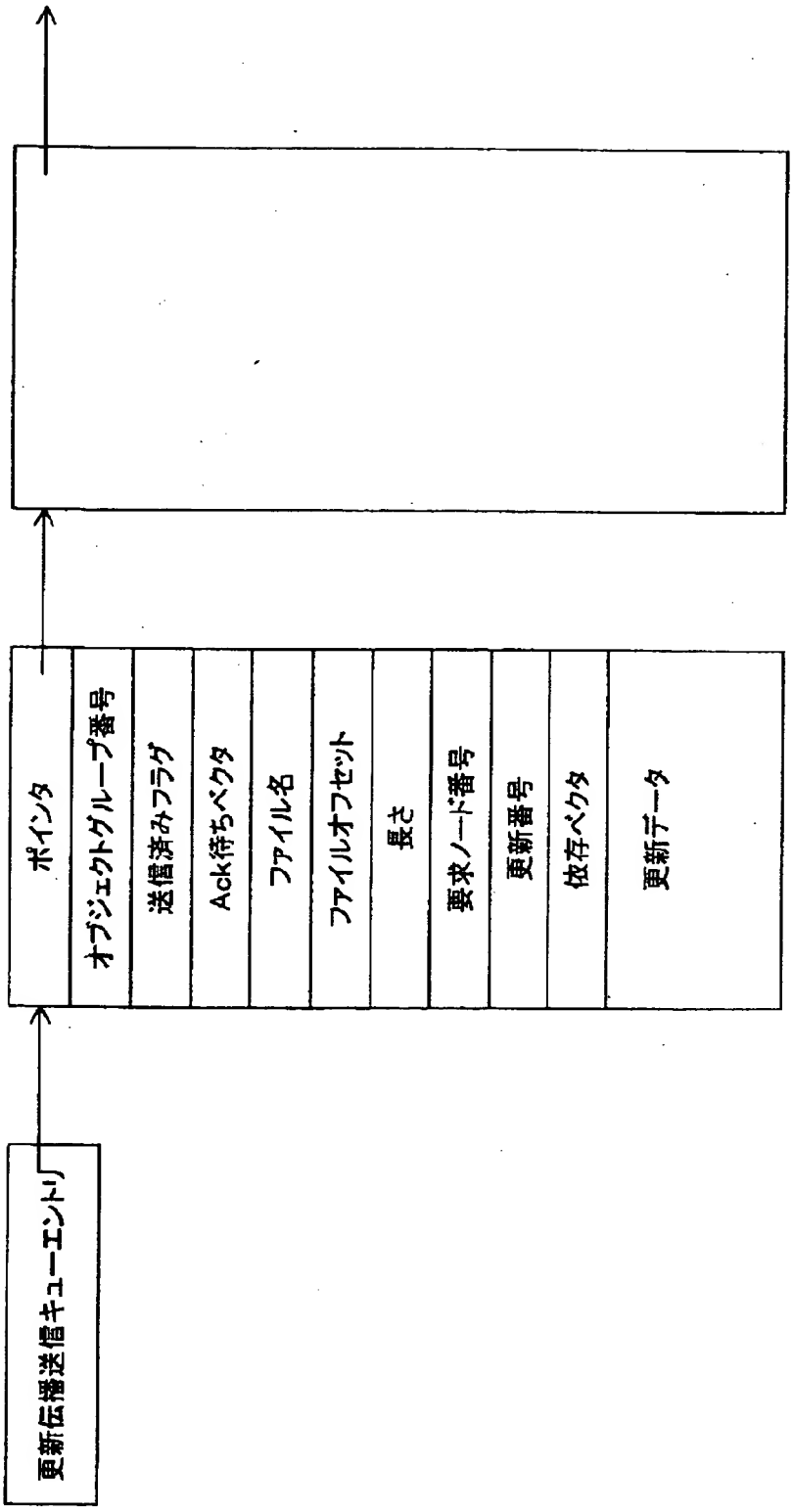
【図２５】

SYNC要求処理時の変更データ通知部の動作処理を示すフローチャート



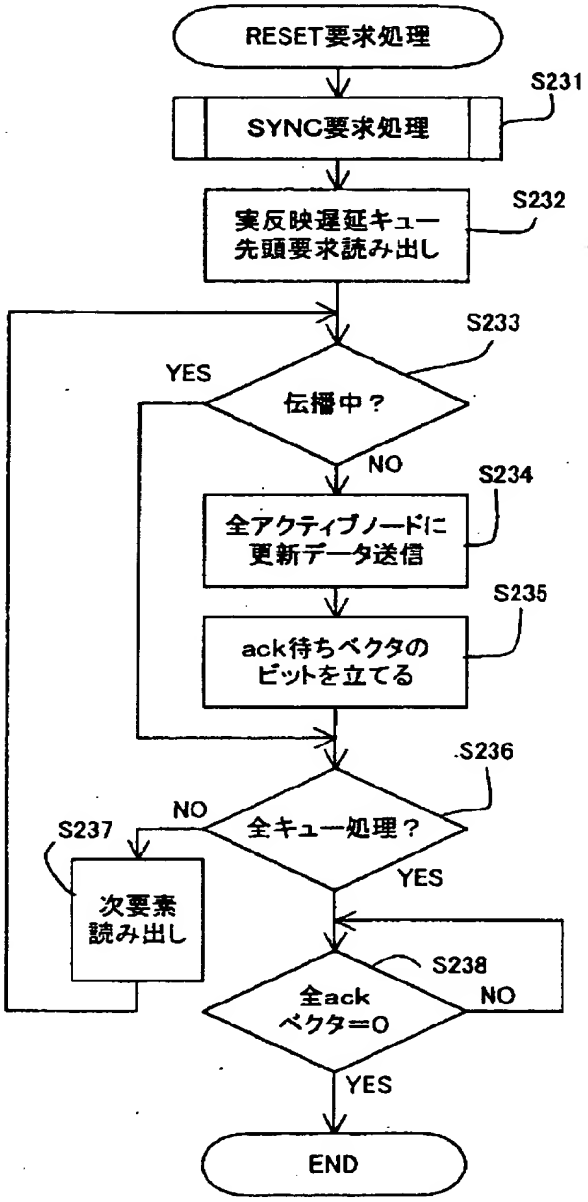
【図２６】

更新伝播送信キューの構成例を示す図



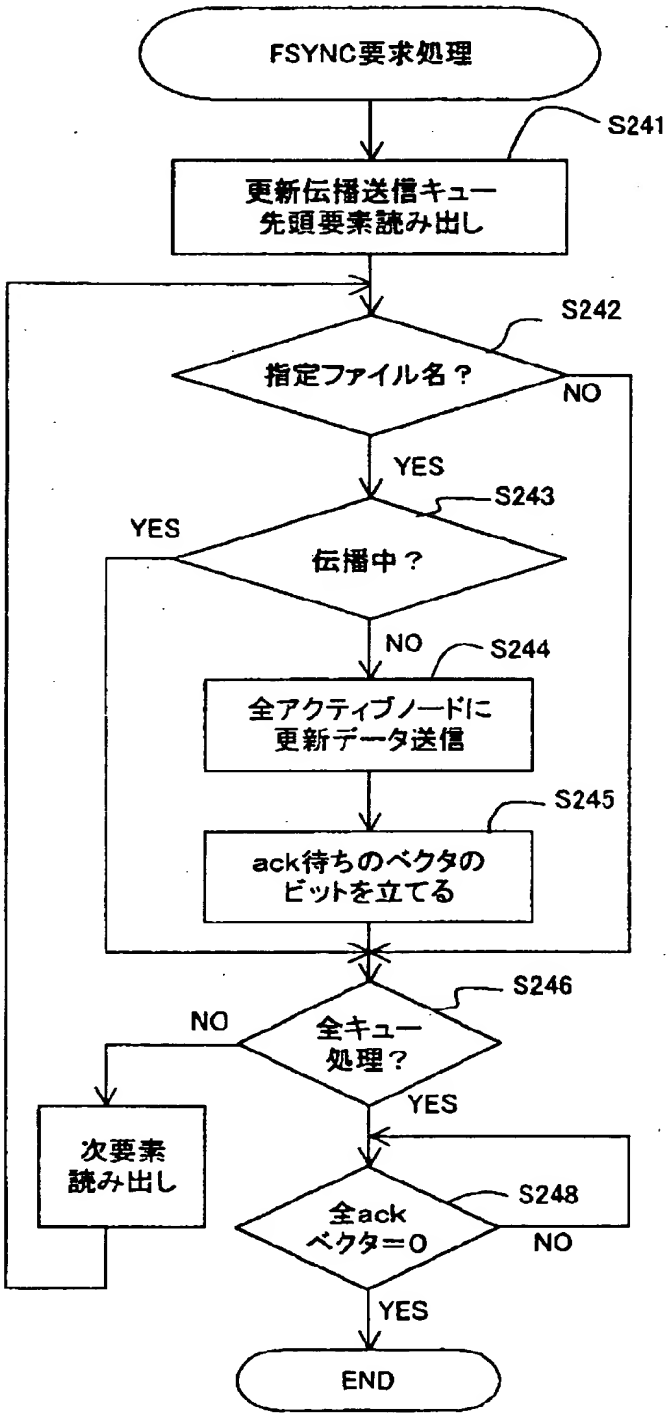
【図２７】

RESET要求処理時の変更データ通知部の動作処理を示すフローチャート



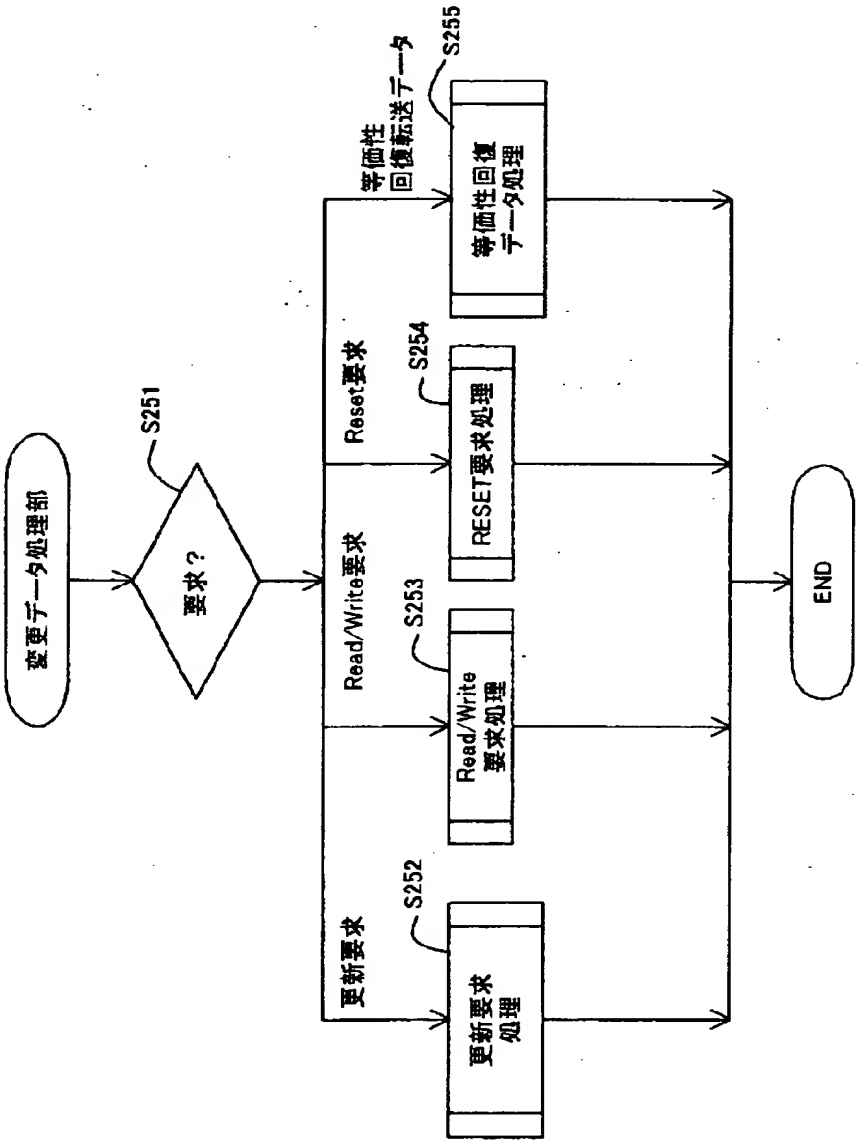
【図２８】

FSYNC要求処理時の変更データ通知部の動作処理を示すフローチャート



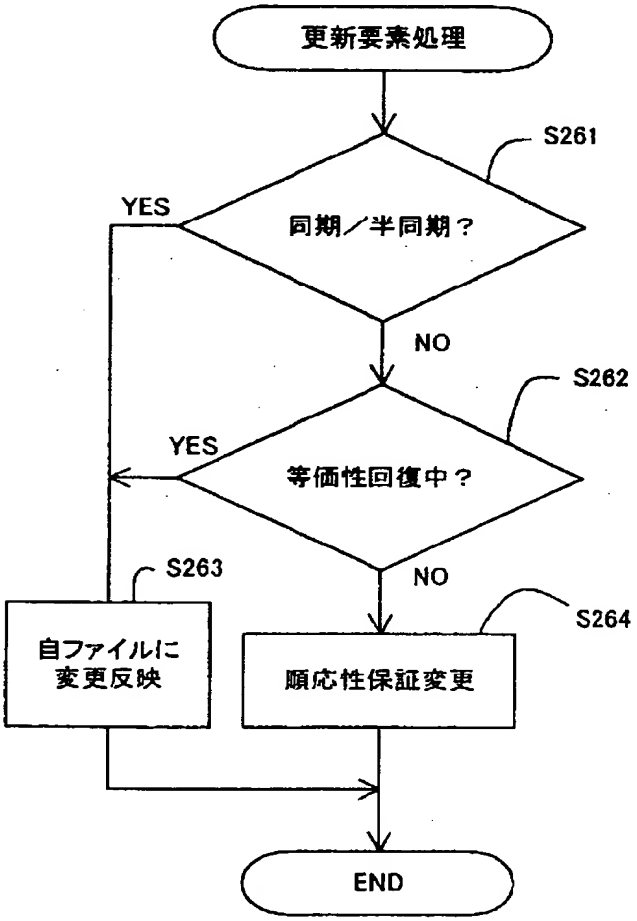
【図２９】

受信データ処理部の動作処理を示すフローチャート



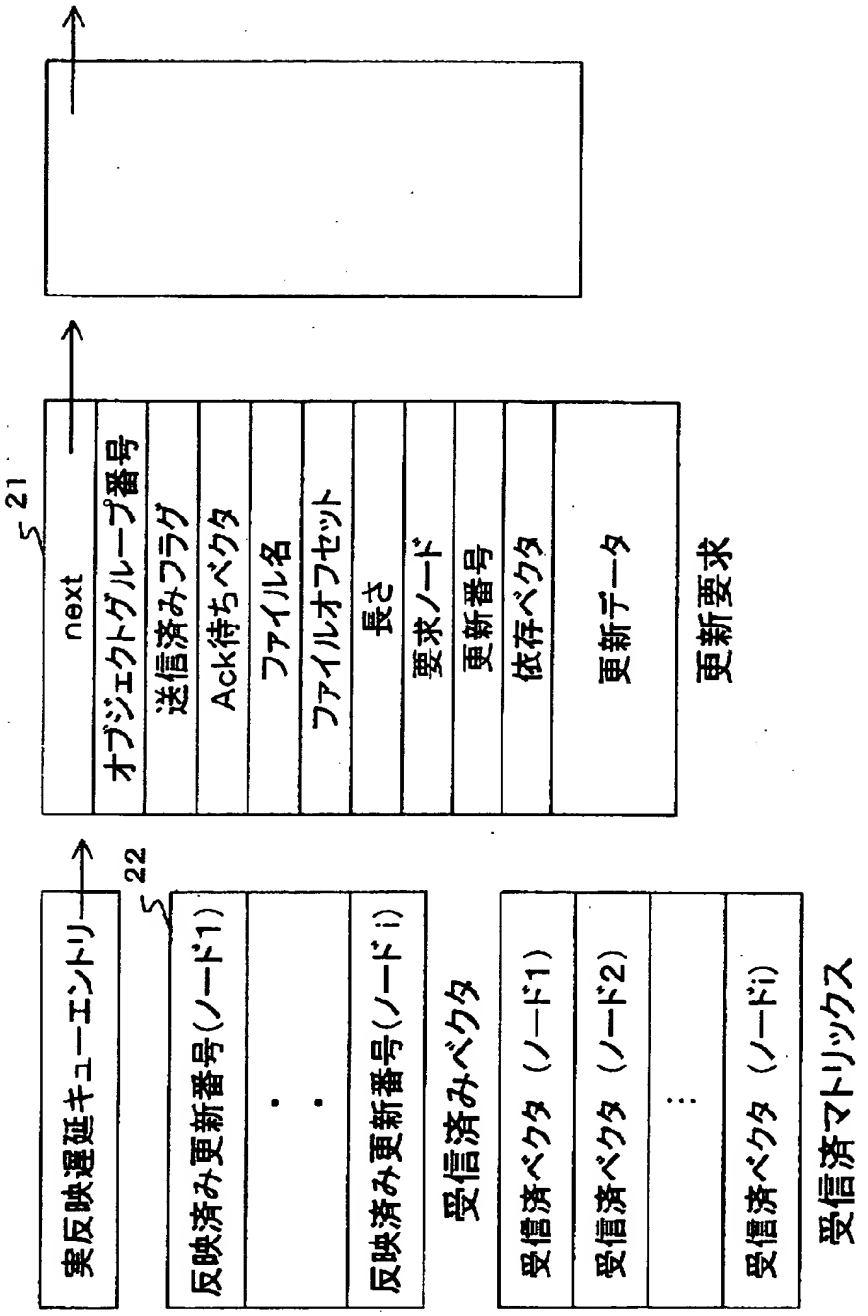
【図３０】

更新要求処理における受信データ処理部の動作処理を示すフローチャート



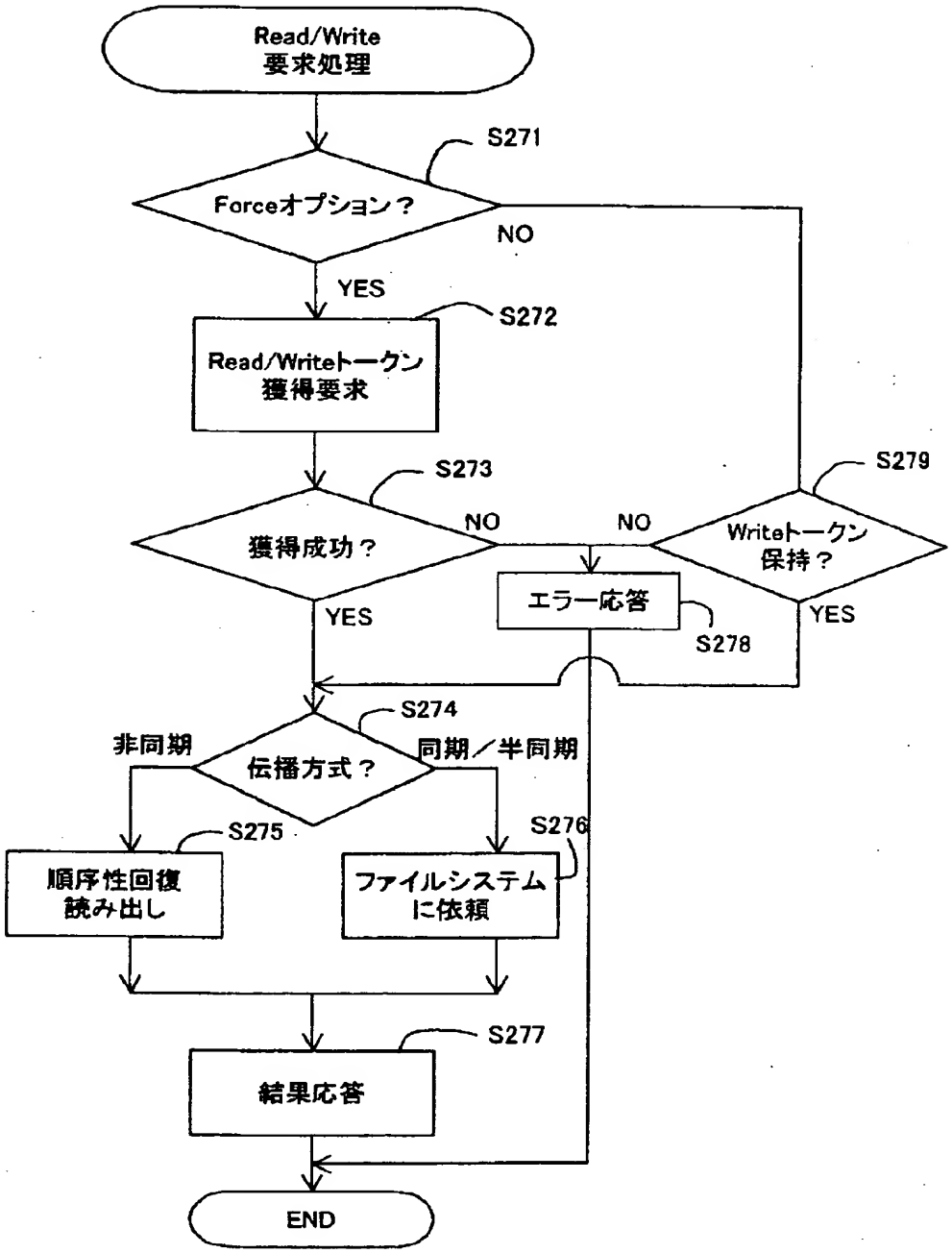
【図３１】

実反映遅延キューの構成例を示す図



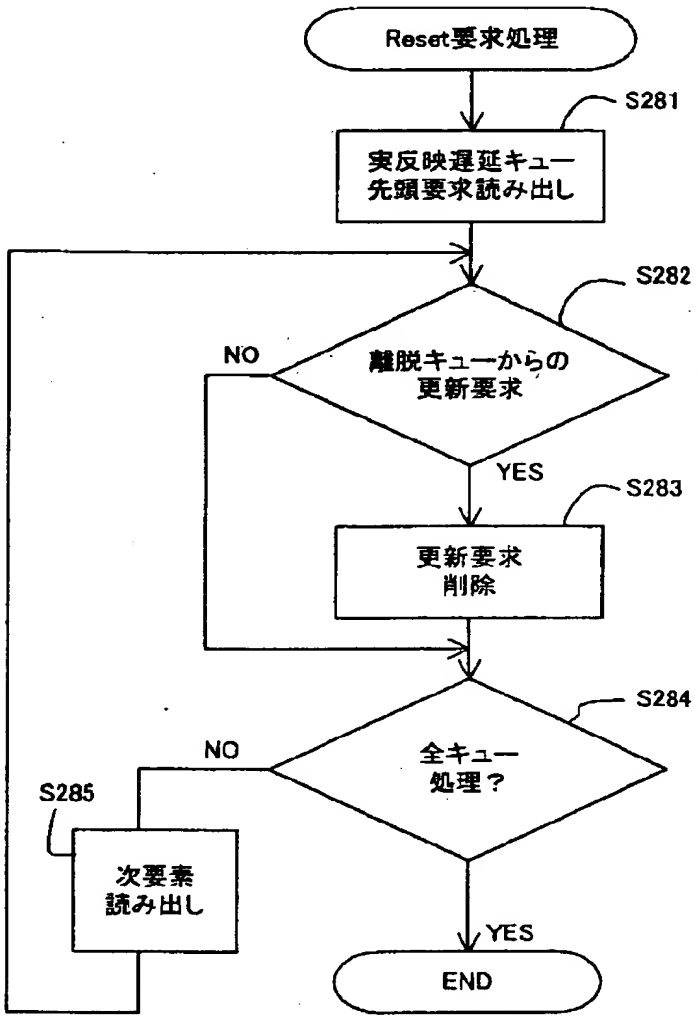
【図３２】

Read/Write要求処理における受信データ処理部の処理を示すフローチャート



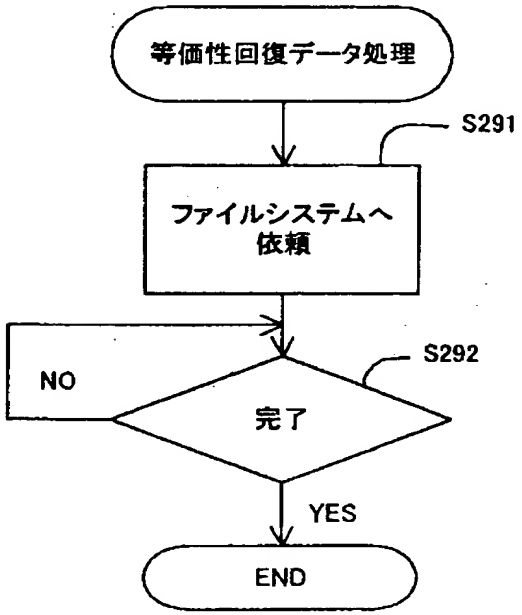
【図３３】

Reset要求処理における受信データ処理部の動作処理を示すフローチャート



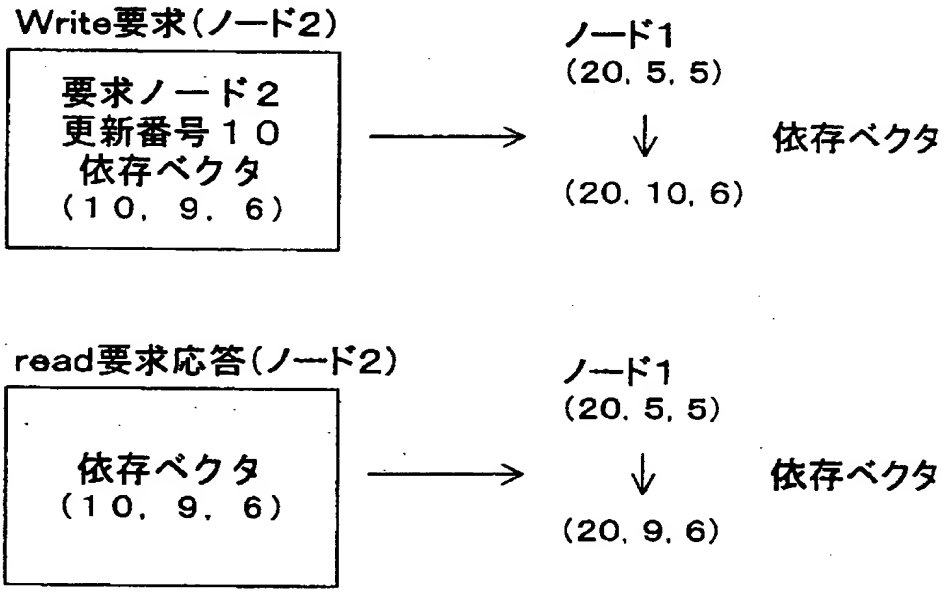
【図３４】

等価性回復データ処理における受信データ処理部の動作処理を示すフローチャート



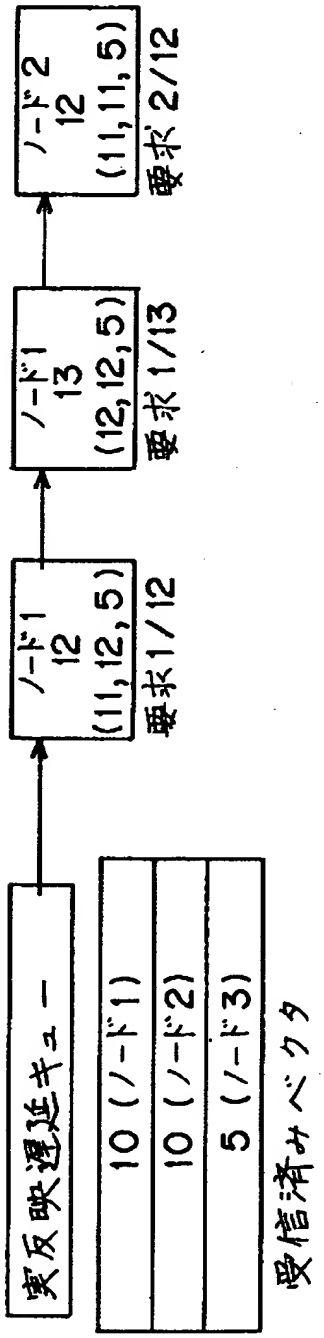
【図３５】

Write要求及びRead要求の応答に付加される
依存ベクタの例を示す図



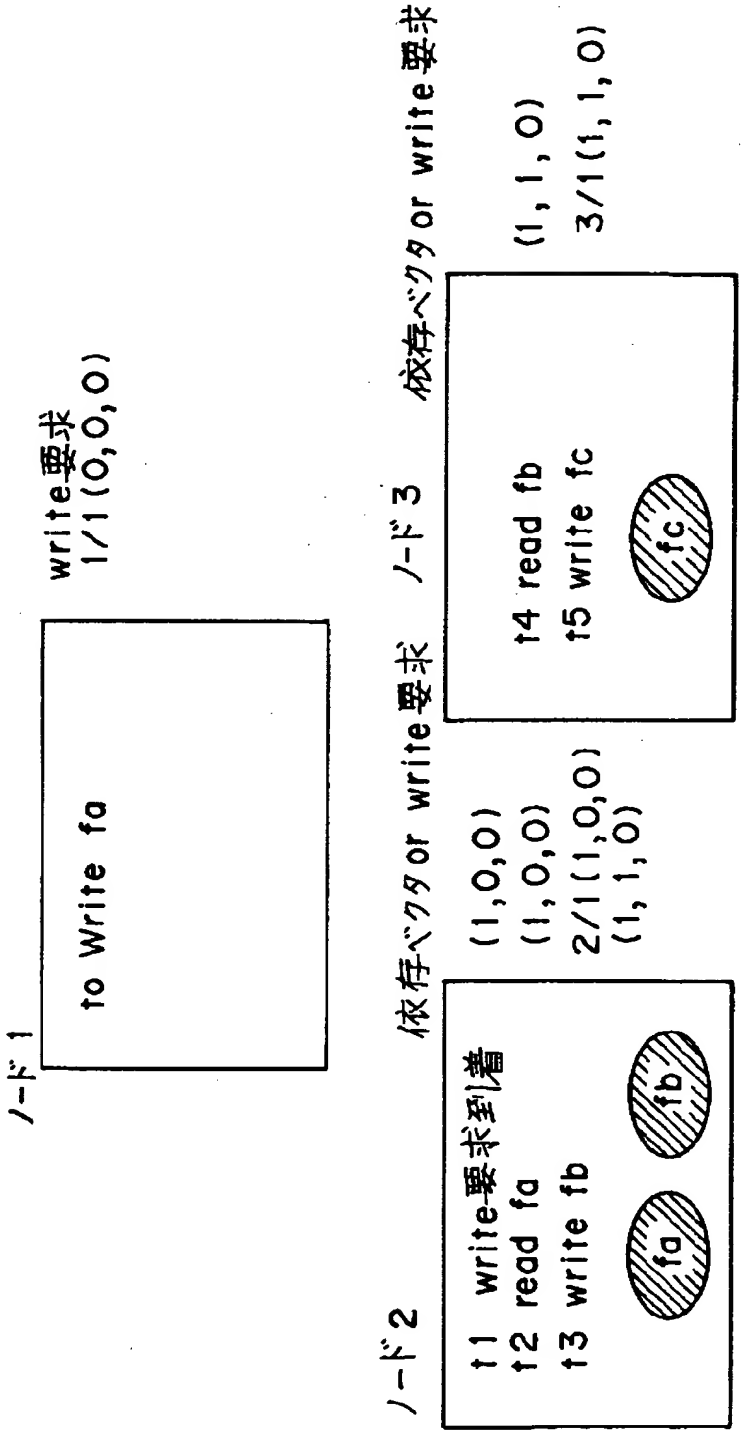
【図３６】

Write要求及びRead要求の応答に
付加される依存ベクタの例を示す図



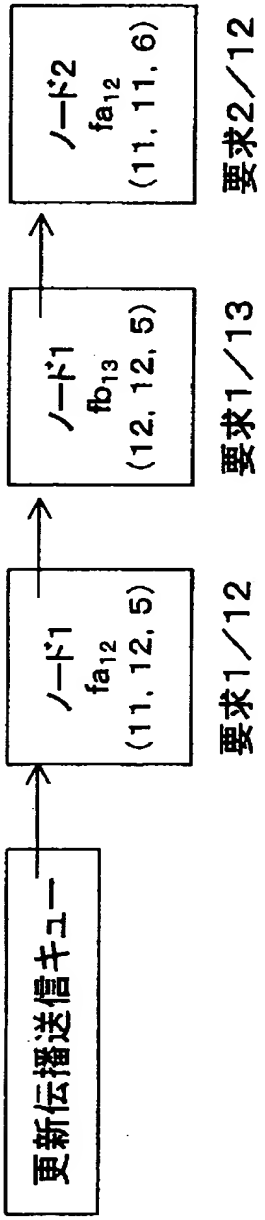
【図３７】

依存関係のある更新要求の順序性の保証を示す図



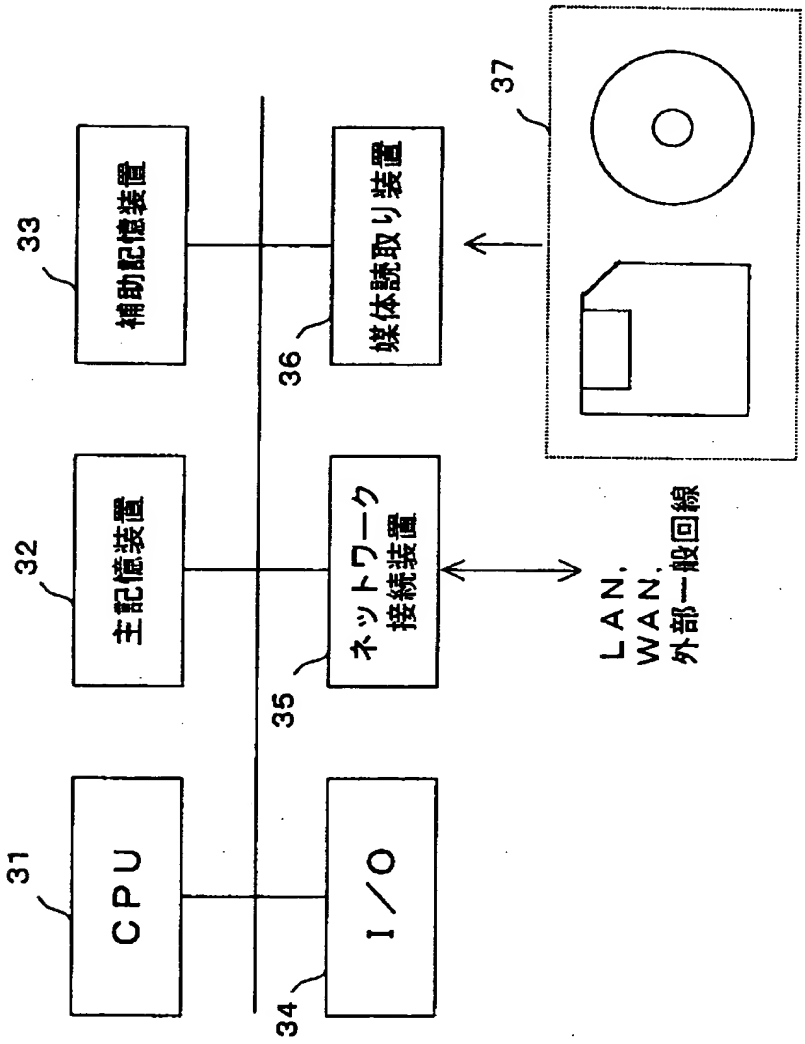
【図３８】

他ノードからのWrite要求を処理する時に、
更新伝播送信キューに同じファイルに対する
更新要求が存在していた場合の処理を説明する図



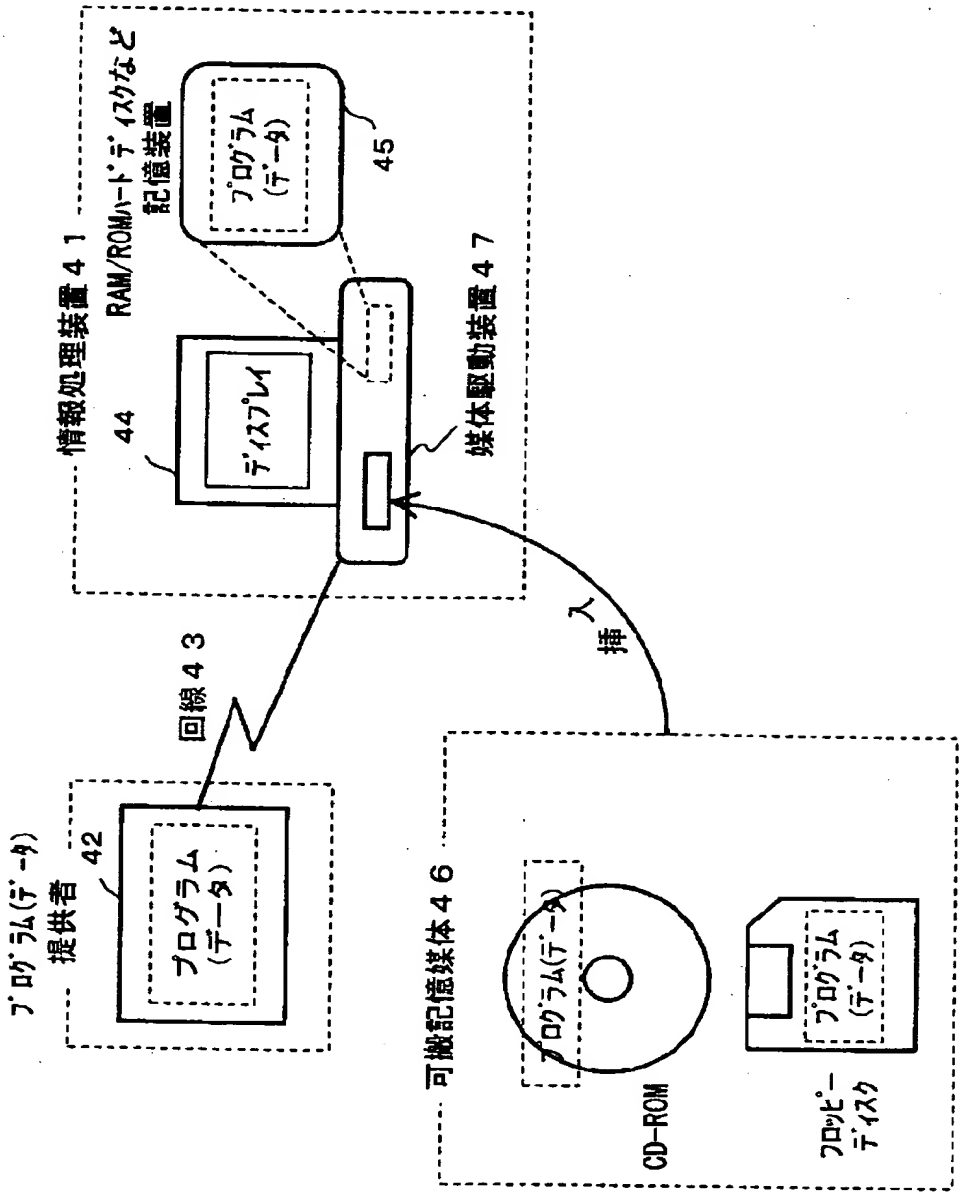
【図３９】

ノードとなる計算機のシステムの環境図



【図４０】

記憶媒体の例を示す図



【書類名】　　要約書

【要約】

【課題】　一貫性保証制御を付加し、ファイルレプリケーションの適用範囲を拡大したファイルレプリケーションシステムを提供することを課題とする。

【解決手段】　自ノード内で生じた共用ファイル６に対するアクセス要求に対し、ＩＯ要求インタセプト手段２はトークン管理手段３に共用ファイル６に対するアクセス許可を求める。トークン管理手段６は、トークン管理手段１３は、ＩＯ要求インタセプト手段２からのアクセス許可要求に対し、既に他のノードが共用ファイルに対する更新許可を保持する時、更新許可を保持するノードをＩＯ要求インタセプト手段２に通知する。ＩＯ要求インタセプト手段は、トークン管理手段からアクセス許可が得られない時、通知された更新許可を保持するノードに共用ファイルへのアクセス処理を依頼する。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　　富士通株式会社